Information, Physics and Computation

Marc Mézard

Laboratoire de Physique Théorique et Modèles Statistiques,
CNRS and Université Paris Sud

Andrea Montanari

Departments of Electrical Engineering and Statistics,
Stanford University

Preface

The following are a few chapters from the book "Information, Physics and Computation" by M. Mézard and Andrea Montanari, collected here for the course given by Marc Mézard at the summer school in Peyresq, July 2008. The book is scheduled for the end of 2008. Please don't circulate the present preliminary version, and don't hesitate to signal us any misprint or error!

The book itself is structured in five large parts, focusing on topics of increasing complexity. Each part typically contains three chapters presenting some core topics in each of the disciplines: information theory, statistical physics, and combinatorial optimization. The topics in each part have a common mathematical structure, which is developed in additional chapters serving as bridges.

- Part A (chapters 1-4) contains introductory chapters to each of the three disciplines and some common probabilistic tools.
- Part B (chapters 5-8) deals with problems in which independence plays an important role: random energy model, random code ensemble, and number partitioning. Thanks to independence of random variables, classical techniques can be successfully to these problems. It ends up with a description of the replica method.
- Part C (chapters 9-13) describes ensembles of problems on graphs: satisfiability, low density parity check codes, and spin glasses. Factor graphs and statistical inference provide a common language.
- Part D (chapters 14-17) explains belief propagation and the related 'replica symmetric' cavity method. These can be thought as approaches to study systems of correlated random variables on large graphs. It shows the success of this approach on three problems: decoding, assignment, and ferromagnets.
- Part E (chapters 18-22) is dedicated to the proliferation of pure states and 'replica symmetry breaking.' It starts with the simpler problem of random linear equations with Boolean variables, then develops the general approach and applies it to satisfiability and coding. The final chapter reviews open problems.

Contents

1
Introduction to Information Theory

This chapter introduces some of the basic concepts of information theory, as well as the definitions and notations of probability theory that will be used throughout the book. The notion of entropy, which is fundamental to the whole topic of this book, is introduced here. We also present the main questions of information theory, data compression and error correction, and state Shannon's theorems.

Sec. 1.1 introduces basic notations in probability. The notion of entropy, and the entropy rate of a sequence , are discussed in Sec. 1.2 and 1.3. A very important concept in information theory is the mutual information of correlated random variables, is is introduced in 1.4. Then we move to the two main aspects of the theory, the compression of data in Sec. 1.5 and the transmission in Sec. 1.6.

1.1 Random variables

The main object of this book will be the behavior of large sets of **discrete random variables** . A discrete random variable X is completely defined[1] by the set of values it can take, $\mathcal{X}$, which we assume to be a finite set, and its **probability distribution** $\{p_X(x)\}_{x\in\mathcal{X}}$. The value $p_X(x)$ is the probability that the random variable X takes the value x. The probability distribution $p_X : \mathcal{X} \to [0,1]$ is a non-negative function that satisfies the normalization condition:

$$\sum_{x\in\mathcal{X}} p_X(x) = 1 \ . \tag{1.1}$$

We shall denote by $\mathbb{P}(A)$ the probability of an **event** $A \subseteq \mathcal{X}$, so that $p_X(x) = \mathbb{P}(X = x)$. To lighten notations, when there is no ambiguity, we use $p(x)$ to denote $p_X(x)$.

If $f(X)$ is a real valued function of the random variable X, the **expectation value** of $f(X)$, which we shall also call the **average** of f, is denoted by:

$$\mathbb{E}\, f = \sum_{x\in\mathcal{X}} p_X(x) f(x) \ . \tag{1.2}$$

While our main focus will be on random variables taking values in finite spaces, we shall sometimes make use of **continuous random variables** taking values in $\mathbb{R}^d$ or in some smooth finite-dimensional manifold. The probability measure for an 'infinitesimal element' $\mathrm{d}x$ will be denoted by $\mathrm{d}p_X(x)$. Each time p_X admits a density

[1] In probabilistic jargon (which we shall avoid hereafter), we take the probability space $(\mathcal{X}, \mathsf{P}(\mathcal{X}), p_X)$ where $\mathsf{P}(\mathcal{X})$ is the σ-field of the parts of $\mathcal{X}$ and $p_X = \sum_{x\in X} p_X(x)\, \delta_x$.

(with respect to the Lebesgue measure), we shall use the notation $p_X(x)$ for the value of this density at the point x. The total probability $\mathbb{P}(X \in \mathcal{A})$ that the variable X takes value in some (measurable) set $\mathcal{A} \subseteq \mathcal{X}$ is given by the integral:

$$\mathbb{P}(X \in \mathcal{A}) = \int_{x \in \mathcal{A}} \mathrm{d}p_X(x) = \int \mathbb{I}(x \in \mathcal{A})\, \mathrm{d}p_X(x) \ , \tag{1.3}$$

where the second form uses the **indicator function** $\mathbb{I}(s)$ of a logical statement s,which is defined to be equal to 1 if the statement s is true, and equal to 0 if the statement is false.

The expectation value $\mathbb{E}\, f(X)$ and the variance $\mathrm{Var}\, f(X)$ of a real valued function $f(x)$ are given by:

$$\mathbb{E}\, f(X) = \int f(x)\ \mathrm{d}p_X(x) \quad ; \qquad \mathrm{Var}\, f(X) = \mathbb{E}\{f(X)^2\} - \{\mathbb{E}\, f(X)\}^2 \tag{1.4}$$

Sometimes we may write $\mathbb{E}_X f(X)$ for specifying the variable to be integrated over. We shall often use the shorthand **pdf** for the **probability density function** $p_X(x)$.

Example 1.1 A fair dice with M faces has $\mathcal{X} = \{1, 2, \dots, M\}$ and $p(i) = 1/M$ for all $i \in \{1, \dots, M\}$. The average of x is $\mathbb{E}\, X = (1 + \cdots + M)/M = (M+1)/2$.

Example 1.2 Gaussian variable: a continuous variable $X \in \mathbb{R}$ has a Gaussian distribution of mean m and variance σ^2 if its probability density is

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{[x-m]^2}{2\sigma^2}\right) \ . \tag{1.5}$$

One has $\mathbb{E}X = m$ and $\mathbb{E}(X-m)^2 = \sigma^2$.

Appendix A contains definitions and notations for the random variables that we shall encounter most frequently

The notations of this chapter mainly deal with discrete variables. Most of the expressions can be transposed to the case of continuous variables by replacing sums $\sum_x$ by integrals and interpreting $p(x)$ as a probability density.

Exercise 1.1 Jensen's inequality. Let X be a random variable taking value in a set $\mathcal{X} \subseteq \mathbb{R}$ and f a convex function (i.e. a function such that $\forall x, y$ and $\forall \alpha \in [0,1]$: $f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y)$. Then

$$\mathbb{E} f(X) \geq f(\mathbb{E} X) \ . \tag{1.6}$$

Supposing for simplicity that $\mathcal{X}$ is a finite set with $|\mathcal{X}| = n$, prove this equality by recursion on n.

1.2 Entropy

The **entropy** H_X of a discrete random variable X with probability distribution $p(x)$ is defined as

$$H_X \equiv -\sum_{x\in\mathcal{X}} p(x)\log_2 p(x) = \mathbb{E}\log_2\left[\frac{1}{p(X)}\right] , \tag{1.7}$$

where we define by continuity $0\log_2 0 = 0$. We shall also use the notation $H(p)$ whenever we want to stress the dependence of the entropy upon the probability distribution of X.

In this Chapter we use the logarithm to the base 2, which is well adapted to digital communication, and the entropy is then expressed in **bits**. In other contexts, and in particular in statistical physics, one rather uses the natural logarithm (with base $e \approx 2.7182818$). It is sometimes said that, in this case, entropy is measured in **nats**. In fact, the two definitions differ by a global multiplicative constant, which amounts to a change of units. When there is no ambiguity we use H instead of H_X.

Intuitively, the entropy H_X is a measure of the uncertainty of the random variable X. One can think of it as the missing information: the larger the entropy, the less a priori information one has on the value of the random variable. It roughly coincides with the logarithm of the number of typical values that the variable can take, as the following examples show.

Example 1.3 A fair coin has two values with equal probability. Its entropy is 1 bit.

Example 1.4 Imagine throwing M fair coins: the number of all possible outcomes is 2^M. The entropy equals M bits.

Example 1.5 A fair dice with M faces has entropy $\log_2 M$.

Example 1.6 Bernoulli process. A Bernoulli random variable X can take values $0, 1$ with probabilities $p(0) = q$, $p(1) = 1 - q$. Its entropy is

$$H_X = -q\log_2 q - (1-q)\log_2(1-q) , \tag{1.8}$$

it is plotted as a function of q in Fig.1.1. This entropy vanishes when $q = 0$ or $q = 1$ because the outcome is certain, it is maximal at $q = 1/2$ when the uncertainty on the outcome is maximal.

Since Bernoulli variables are ubiquitous, it is convenient to introduce the function $\mathcal{H}(q) \equiv -q\log q - (1-q)\log(1-q)$, for their entropy.

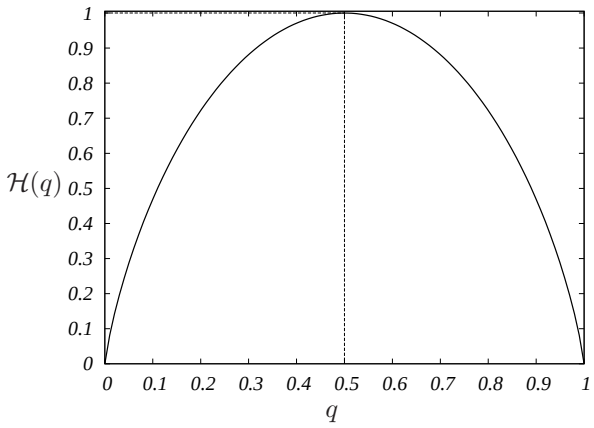

Fig. 1.1 The entropy $\mathcal{H}(q)$ of a binary variable with $p(X=0)=q$, $p(X=1)=1-q$, plotted versus q

Exercise 1.2 An unfair dice with four faces and $p(1)=1/2,\ p(2)=1/4,\ p(3)=p(4)=1/8$ has entropy $H=7/4$, smaller than the one of the corresponding fair dice.

Exercise 1.3 DNA is built from a sequence of bases which are of four types, A,T,G,C. In natural DNA of primates, the four bases have nearly the same frequency, and the entropy per base, if one makes the simplifying assumptions of independence of the various bases, is $H=-\log_2(1/4)=2$. In some genus of bacteria, one can have big differences in concentrations: $p(G)=p(C)=0.38$, $p(A)=p(T)=0.12$, giving a smaller entropy $H\approx 1.79$.

Exercise 1.4 In some intuitive way, the entropy of a random variable is related to the 'risk' or 'surprise' which are associated to it. Let us see how these notions can be made more precise.

Consider a gambler who bets on a sequence of Bernoulli random variables $X_t\in\{0,1\}$, $t\in\{0,1,2,\dots\}$ with mean $\mathbb{E}X_t=p$. Imagine he knows the distribution of the X_t's and, at time t, he bets a fraction $w(1)=p$ of his money on 1 and a fraction $w(0)=(1-p)$ on 0. He looses whatever is put on the wrong number, while he doubles whatever has been put on the right one. Define the average doubling rate of his wealth at time t as

$$W_t=\frac{1}{t}\,\mathbb{E}\,\log_2\left\{\prod_{t'=1}^{t}2w(X_{t'})\right\}\,. \tag{1.9}$$

It is easy to prove that the expected doubling rate $\mathbb{E}W_t$ is related to the entropy of X_t: $\mathbb{E}W_t=1-\mathcal{H}(p)$. In other words, it is easier to make money out of predictable events.

Another notion that is directly related to entropy is the **Kullback-Leibler (KL) divergence** between two probability distributions $p(x)$ and $q(x)$ over the same finite space $\mathcal{X}$. This is defined as:

$$D(q||p) \equiv \sum_{x\in\mathcal{X}} q(x) \log \frac{q(x)}{p(x)} \tag{1.10}$$

where we adopt the conventions $0 \log 0 = 0$, $0 \log(0/0) = 0$. It is easy to show that: (i) $D(q||p)$ is convex in $q(x)$; (ii) $D(q||p) \geq 0$; (iii) $D(q||p) > 0$ unless $q(x) \equiv p(x)$. The last two properties derive from the concavity of the logarithm (i.e. the fact that the function $-\log x$ is convex) and Jensen's inequality (1.6): if $\mathbb{E}$ denotes expectation with respect to the distribution $q(x)$, then $-D(q||p) = \mathbb{E} \log[p(x)/q(x)] \leq \log \mathbb{E}[p(x)/q(x)] = 0$. The KL divergence $D(q||p)$ thus looks like a distance between the probability distributions q and p, although it is not symmetric.

The importance of the entropy, and its use as a measure of information, derives from the following properties:

1. $H_X \geq 0$.
2. $H_X = 0$ if and only if the random variable X is certain, which means that X takes one value with probability one.
3. Among all probability distributions on a set $\mathcal{X}$ with M elements, H is maximum when all events x are equiprobable, with $p(x) = 1/M$. The entropy is then $H_X = \log_2 M$.
   To prove this statement, notice that if $\mathcal{X}$ has M elements then the KL divergence $D(p||\overline{p})$ between $p(x)$ and the uniform distribution $\overline{p}(x) = 1/M$ is $D(p||\overline{p}) = \log_2 M - H(p)$. The statement is a direct consequence of the properties of the KL divergence.
4. If X and Y are two **independent** random variables, meaning that $p_{X,Y}(x, y) = p_X(x)p_Y(y)$, the total entropy of the pair X, Y is equal to $H_X + H_Y$:

$$\begin{aligned} H_{X,Y} &= -\sum_{x,y} p_{X,Y}(x, y) \log_2 p_{X,Y}(x, y) \\ &= -\sum_{x,y} p_{X,Y}(x, y) \left(\log_2 p_X(x) + \log_2 p_Y(y)\right) = H_X + H_Y \end{aligned} \tag{1.11}$$

5. For any pair of random variables, one has in general $H_{X,Y} \leq H_X + H_Y$, and this result is immediately generalizable to n variables. (The proof can be obtained by using the positivity of KL divergence $D(p_1||p_2)$, where $p_1 = p_{X,Y}$ and $p_2 = p_X p_Y$).
6. Additivity for composite events. Take a finite set of events $\mathcal{X}$, and decompose it into $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2$, where $\mathcal{X}_1 \cap \mathcal{X}_2 = \emptyset$. Call $q_1 = \sum_{x\in\mathcal{X}_1} p(x)$ the probability of $\mathcal{X}_1$, and q_2 the probability of $\mathcal{X}_2$. For each $x \in \mathcal{X}_1$, define as usual the conditional probability of x, given that $x \in \mathcal{X}_1$, by $r_1(x) = p(x)/q_1$ and define similarly $r_2(x)$ as the conditional probability of x, given that $x \in \mathcal{X}_2$. Then the total entropy can be written as the sum of two contributions $H_X = -\sum_{x\in\mathcal{X}} p(x) \log_2 p(x) = H(q) + \widetilde{H}(q, r)$, where:

$$H(q) = -q_1 \log_2 q_1 - q_2 \log_2 q_2 \tag{1.12}$$

$$\widetilde{H}(q,r) = -q_1 \sum_{x\in\mathcal{X}_1} r_1(x) \log_2 r_1(x) - q_2 \sum_{x\in\mathcal{X}_1} r_2(x) \log_2 r_2(x) \tag{1.13}$$

The proof is straightforward by substituting the laws r_1 and r_2 by their definitions. This property is interpreted as the fact that the average information associated to the choice of an event x is additive, being the sum of the relative information $H(q)$ associated to a choice of subset, and the information $H(r)$ associated to the choice of the event inside the subsets (weighted by the probability of the subsets). It is the main property of the entropy, which justifies its use as a measure of information. In fact, this is a simple example of the so called chain rule for conditional entropy, which will be further illustrated in Sec. 1.4.

Conversely, these properties together with appropriate hypotheses of continuity and monotonicity can be used to define axiomatically the entropy.

1.3 Sequences of random variables and entropy rate

In many situations of interest one deals with a random process which generates **sequences of random variables** $\{X_t\}_{t\in\mathbb{N}}$, each of them taking values in the same finite space $\mathcal{X}$. We denote by $P_N(x_1,\dots,x_N)$ the joint probability distribution of the first N variables. If $A \subset \{1,\dots,N\}$ is a subset of indices, we shall denote by $\overline{A}$ its complement $\overline{A} = \{1,\dots,N\} \setminus A$ and use the notations $\underline{x}_A = \{x_i, i \in A\}$ and $\underline{x}_{\overline{A}} = \{x_i, i \in \overline{A}\}$ (the set subscript will be dropped whenever clear from the context). The **marginal distribution** of the variables in A is obtained by summing P_N on the variables in $\overline{A}$:

$$P_A(\underline{x}_A) = \sum_{\underline{x}_{\overline{A}}} P_N(x_1,\dots,x_N)\ . \tag{1.14}$$

Example 1.7 The simplest case is when the X_t's are independent. This means that $P_N(x_1,\dots,x_N) = p_1(x_1)p_2(x_2)\dots p_N(x_N)$. If all the distributions p_i are identical, equal to p, the variables are **independent identically distributed**, which will be abbreviated as **i.i.d.**. The joint distribution is

$$P_N(x_1,\dots,x_N) = \prod_{t=1}^{N} p(x_i)\,. \tag{1.15}$$

Example 1.8 The sequence $\{X_t\}_{t\in\mathbb{N}}$ is said to be a **Markov chain** if

$$P_N(x_1,\dots,x_N) = p_1(x_1)\prod_{t=1}^{N-1} w(x_t\to x_{t+1})\,. \tag{1.16}$$

Here $\{p_1(x)\}_{x\in\mathcal{X}}$ is called the **initial state**, and $\{w(x\to y)\}_{x,y\in\mathcal{X}}$ are the **transition probabilities** of the chain. The transition probabilities must be non-negative and normalized:

$$\sum_{y\in\mathcal{X}} w(x\to y) = 1\,, \qquad \text{for any } x\in\mathcal{X}. \tag{1.17}$$

When we have a sequence of random variables generated by a certain process, it is intuitively clear that the entropy grows with the number N of variables. This intuition suggests to define the **entropy rate** of a sequence $\underline{x}_N \equiv \{X_t\}_{t\in\mathbb{N}}$ as

$$h_X = \lim_{N\to\infty} H_{\underline{X}_N}/N\,, \tag{1.18}$$

if the limit exists. The following examples should convince the reader that the above definition is meaningful.

Example 1.9 If the X_t's are i.i.d. random variables with distribution $\{p(x)\}_{x\in\mathcal{X}}$, the additivity of entropy implies

$$h_X = H(p) = -\sum_{x\in\mathcal{X}} p(x)\log p(x)\,. \tag{1.19}$$

Example 1.10 Let $\{X_t\}_{t\in\mathbb{N}}$ be a Markov chain with initial state $\{p_1(x)\}_{x\in\mathcal{X}}$ and transition probabilities $\{w(x\to y)\}_{x,y\in\mathcal{X}}$. Call $\{p_t(x)\}_{x\in\mathcal{X}}$ the marginal distribution of X_t and assume the following limit to exist independently of the initial condition:

$$p^*(x) = \lim_{t\to\infty} p_t(x)\,. \tag{1.20}$$

As we shall see in chapter **??**, this turns indeed to be true under quite mild hypotheses on the transition probabilities $\{w(x\to y)\}_{x,y\in\mathcal{X}}$. Then it is easy to show that

$$h_X = -\sum_{x,y\in\mathcal{X}} p^*(x)\,w(x\to y)\log w(x\to y)\,. \tag{1.21}$$

If you imagine for instance that a text in English is generated by picking letters randomly in the alphabet $\mathcal{X}$, with empirically determined transition probabilities $w(x\to y)$, then Eq. (1.21) gives a rough estimate of the entropy of English.

A more realistic model is obtained using a Markov chain *with memory*. This means that each new letter x_{t+1} depends on the past through the value of the k previous letters $x_t, x_{t-1}, \dots, x_{t-k+1}$. Its conditional distribution is given by the transition probabilities $w(x_t, x_{t-1}, \dots, x_{t-k+1} \to x_{t+1})$. Computing the corresponding entropy rate is easy. For $k = 4$ one gets an entropy of 2.8 bits per letter, much smaller than the trivial upper bound $\log_2 27$ (there are 26 letters, plus the space symbols), but many words so generated are still not correct English words. Better estimates of the entropy of English, through guessing experiments, give a number around 1.3.

1.4 Correlated variables and mutual information

Given two random variables X and Y, taking values in $\mathcal{X}$ and $\mathcal{Y}$, we denote their joint probability distribution as $p_{X,Y}(x,y)$, which is abbreviated as $p(x,y)$, and the conditional probability distribution for the variable y given x as $p_{Y|X}(y|x)$, abbreviated as $p(y|x)$. The reader should be familiar with Bayes classical theorem:

$$p(y|x) = p(x,y)/p(x)\ . \tag{1.22}$$

When the random variables X and Y are independent, $p(y|x)$ is x-independent. When the variables are dependent, it is interesting to have a measure on their degree of dependence: how much information does one obtain on the value of y if one knows x? The notions of conditional entropy and mutual information will answer this question.

Let us define the **conditional entropy** $H_{Y|X}$ as the entropy of the law $p(y|x)$, averaged over x:

$$H_{Y|X} \equiv -\sum_{x\in\mathcal{X}} p(x) \sum_{y\in\mathcal{Y}} p(y|x)\log_2 p(y|x)\,. \tag{1.23}$$

The joint entropy $H_{X,Y} \equiv -\sum_{x\in\mathcal{X},y\in\mathcal{Y}} p(x,y)\log_2 p(x,y)$ of the pair of variables x, y can be written as the entropy of x plus the conditional entropy of y given x, an identity known as the **chain rule**:

$$H_{X,Y} = H_X + H_{Y|X}\,. \tag{1.24}$$

In the simple case where the two variables are independent, $H_{Y|X} = H_Y$, and $H_{X,Y} = H_X + H_Y$. One way to measure the correlation of the two variables is the **mutual information** $I_{X,Y}$ which is defined as:

$$I_{X,Y} \equiv \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)} \ . \tag{1.25}$$

It is related to the conditional entropies by:

$$I_{X,Y} = H_Y - H_{Y|X} = H_X - H_{X|Y} \ . \tag{1.26}$$

This shows that the mutual information $I_{X,Y}$ measures the reduction in the uncertainty of x due to the knowledge of y, and is symmetric in x, y.

Proposition 1.11 *$I_{X,Y} \geq 0$. Moreover $I_{X,Y} = 0$ if and only if X and Y are independent variables.*

Proof: Write $I_{X,Y} = \mathbb{E}_{x,y} - \log_2 \frac{p(x)p(y)}{p(x,y)}$. Consider the random variable $u = (x, y)$ with probability distribution $p(x, y)$. As the function $-\log(\,\cdot\,)$ is convex, one can apply Jensen's inequality (1.6). This gives the result $I_{X,Y} \geq 0$ □

Exercise 1.5 A large group of friends plays the following game (telephone without cables). The guy number zero chooses a number $X_0 \in \{0, 1\}$ with equal probability and communicates it to the first one without letting the others hear, and so on. The first guy communicates the number to the second one, without letting anyone else hear. Call X_n the number communicated from the n-th to the $(n + 1)$-th guy. Assume that, at each step a guy gets confused and communicates the wrong number with probability p. How much information does the n-th person have about the choice of the first one?

We can quantify this information through $I_{X_0,X_n} \equiv I_n$. Show that $I_n = 1 - \mathcal{H}(p_n)$ with p_n given by $1 - 2p_n = (1 - 2p)^n$. In particular, as $n \to \infty$

$$I_n = \frac{(1-2p)^{2n}}{2\log 2} \left[1 + O((1-2p)^{2n})\right] \ . \tag{1.27}$$

The 'knowledge' about the original choice decreases exponentially along the chain.

mutual information gets degraded when data is transmitted or processed. This is quantified by:

Proposition 1.12 (Data processing inequality). *Consider a Markov chain $X \to Y \to Z$ (so that the joint probability of the three variables can be written as $p_1(x)w_2(x \to y)w_3(y \to z)$). Then: $I_{X,Z} \leq I_{X,Y}$. In particular, if we apply this result to the case where Z is a function of Y, $Z = f(Y)$, we find that applying f degrades the information: $I_{X,f(Y)} \leq I_{X,Y}$.*

Proof: Let us introduce, in general, the mutual information of two variables conditioned to a third one: $I_{X,Y|Z} = H_{X|Z} - H_{X|(YZ)}$. The mutual information between a variable X and a pair of variables (YZ) can be decomposed using the following

chain rule : $I_{X,(YZ)} = I_{X,Z} + I_{X,Y|Z} = I_{X,Y} + I_{X,Z|Y}$. If we have a Markov chain $X \to Y \to Z$, X and Z are independent when one conditions on the value of Y, therefore $I_{X,Z|Y} = 0$. The result follows from the fact that $I_{X,Y|Z} \geq 0$. □

conditional entropy also gives a bound on the possibility to guess a variable. Suppose you want to guess the value of the random variable X, but you observe only the random variable Y (which can be thought as a noisy version of X). From Y, you compute a function $\widehat{X} = g(Y)$ which is your estimate for X. What is the probability $\mathrm{P_e}$ that you guessed incorrectly? Intuitively, if X and Y are strongly correlated one can expect that $\mathrm{P_e}$ is small, while it increases for less correlated variables. This is quantified by:

Proposition 1.13 (Fano's inequality). *Consider a random variable X taking values in the alphabet $\mathcal{X}$, and the Markov chain $X \to Y \to \widehat{X}$ where $\widehat{X} = g(Y)$ is an estimate for the value of X. Define as $\mathrm{P_e} = \mathbb{P}(\widehat{X} \neq X)$ the probability to make a wrong guess. It is bounded below as follows:*

$$\mathcal{H}(\mathrm{P_e}) + \mathrm{P_e} \log_2(|\mathcal{X}| - 1) \geq H(X|Y)\,. \tag{1.28}$$

Proof: Define the random variable $E = \mathbb{I}(\widehat{X} \neq X)$ equal to 0 if $\widehat{X} = X$, and to 1 otherwise, and decompose the conditional entropy $H_{X,E|Y}$ using the chain rule, in two ways: $H_{X,E|Y} = H_{X|Y} + H_{E|X,Y} = H_{E|Y} + H_{X|E,Y}$. Then notice that: (*i*) $H_{E|X,Y} = 0$ (because E is a function of X and Y); (*ii*) $H_{E|Y} \leq H_E = \mathcal{H}(\mathrm{P_e})$; (*iii*) $H_{X|E,Y} = (1 - \mathrm{P_e})H_{X|E=0,Y} + \mathrm{P_e} H_{X|E=1,Y} = \mathrm{P_e} H_{X|E=1,Y} \leq \mathrm{P_e} \log_2(|\mathcal{X}| - 1)$. □

Exercise 1.6 Suppose that X can take k values, and its distribution is $p(1) = 1 - p$, $p(x) = \frac{p}{k-1}$ for $x \geq 2$. If X and Y are independent, what is the value of the right hand side of Fano's inequality? Assuming that $1 - p > \frac{p}{k-1}$, what is the best guess one can make on the value of X? What is the probability of error? Show that Fano's inequality holds as an equality in this case.

1.5 Data compression

Imagine an information source which generates a sequence of symbols $\underline{X} = \{X_1, \ldots, X_N\}$ taking values in a finite alphabet $\mathcal{X}$. Let us assume a probabilistic model for the source, meaning that the X_i's are random variables. We want to store the information contained in a given realization $\underline{x} = \{x_1 \ldots x_N\}$ of the source in the most compact way.

This is the basic problem of **source coding**. Apart from being an issue of utmost practical interest, it is a very instructive subject. It allows in fact to formalize in a concrete fashion the intuitions of 'information' and 'uncertainty' which are associated with the definition of entropy. Since entropy will play a crucial role throughout the book, we present here a little detour into source coding.

1.5.1 Codewords

We first need to formalize what is meant by "storing the information". We define a **source code** for the random variable $\underline{X}$ to be a mapping w which associates to any possible information sequence in $\mathcal{X}^N$ a string in a reference alphabet which we shall assume to be $\{0,1\}$:

$$\begin{aligned} w : \mathcal{X}^N &\to \{0,1\}^* \\ \underline{x} &\mapsto w(\underline{x})\,. \end{aligned} \tag{1.29}$$

Here we used the convention of denoting by $\{0,1\}^*$ the set of binary strings of arbitrary length. Any binary string which is in the image of w is called a **codeword**.

Often the sequence of symbols $X_1 \dots X_N$ is a part of a longer stream. The compression of this stream is realized in three steps. First the stream is broken into blocks of length N. Then each block is encoded separately using w. Finally the codewords are glued to form a new (hopefully more compact) stream. If the original stream consisted in the blocks $\underline{x}^{(1)}, \underline{x}^{(2)}, \dots, \underline{x}^{(r)}$, the output of the encoding process will be the concatenation of $w(\underline{x}^{(1)}), \dots, w(\underline{x}^{(r)})$. In general there is more than one way of parsing this concatenation into codewords, which may cause troubles when one wants to recover the compressed data. We shall therefore require the code w to be such that any concatenation of codewords can be parsed unambiguously. The mappings w satisfying this property are called **uniquely decodable codes**.

Unique decodability is surely satisfied if for any $\underline{x}, \underline{x}' \in \mathcal{X}^N$, $w(\underline{x})$ is not a prefix of $w(\underline{x}')$ (see Fig. 1.2). In such a case the code is said to be **instantaneous**. Hereafter we shall focus on instantaneous codes, since they are both practical and slightly simpler to analyze.

Now that we precised how to store information, namely using a source code, it is useful to introduce some figure of merit for source codes. If $l_w(x)$ is the length of the string $w(x)$, the average length of the code is:

$$L(w) = \sum_{\underline{x} \in \mathcal{X}^N} p(\underline{x})\, l_w(\underline{x})\ . \tag{1.30}$$

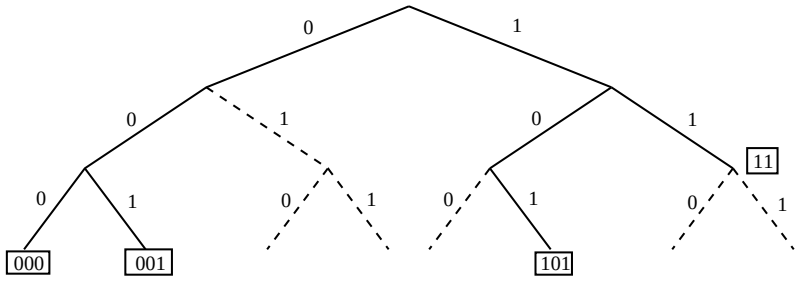

Fig. 1.2 An instantaneous source code: each codeword is assigned to a node in a binary tree in such a way that no one among them is the ancestor of another one. Here the four codewords are framed.

Example 1.14 Take $N = 1$ and consider a random variable X which takes values in $\mathcal{X} = \{1, 2, \ldots, 8\}$ with probabilities $p(1) = 1/2,\ p(2) = 1/4,\ p(3) = 1/8,\ p(4) = 1/16,\ p(5) = 1/32,\ p(6) = 1/64,\ p(7) = 1/128,\ p(8) = 1/128$. Consider the two codes w_1 and w_2 defined by the table below

x	$p(x)$	$w_1(x)$	$w_2(x)$
1	1/2	000	0
2	1/4	001	10
3	1/8	010	110
4	1/16	011	1110
5	1/32	100	11110
6	1/64	101	111110
7	1/128	110	1111110
8	1/128	111	11111110

(1.31)

These two codes are instantaneous. For instance looking at the code w_2, the encoded string 10001101110010 can be parsed in only one way since each symbol 0 ends a codeword . It thus corresponds to the sequence $x_1 = 2, x_2 = 1, x_3 = 1, x_4 = 3, x_5 = 4, x_6 = 1, x_7 = 2$. The average length of code w_1 is $L(w_1) = 3$, the average length of code w_2 is $L(w_2) = 247/128$. Notice that w_2 achieves a shorter average length because it assigns the shortest codeword (namely 0) to the most probable symbol (i.e. $x = 1$).

Example 1.15 A useful graphical representation of a source code is obtained by drawing a binary tree and associating each codeword to the corresponding node in the tree. In Fig. 1.2 we represent in this way a source code with $|\mathcal{X}^N| = 4$. It is quite easy to recognize that the code is indeed instantaneous. The codewords, which are framed, are such that no codeword is the ancestor of any other codeword in the tree. Given a sequence of codewords, parsing is immediate. For instance the sequence 00111000101001 can be parsed only in 001, 11, 000, 101, 001

1.5.2 Optimal compression and entropy

Suppose to have a 'complete probabilistic characterization' of the source you want to compress. What is the 'best code' w for this source?

This problem was solved (to a large extent) by Shannon in his celebrated 1948 paper, by connecting the best achievable average length to the entropy of the source. Following Shannon we assume to know the probability distribution of the source $p(\underline{x})$. Moreover we interpret 'best code' as 'code with the shortest average length'.

Theorem 1.16 *Let L_N^* be the shortest average length achievable by an instantaneous code for the variable $\underline{X} = \{X_1, \dots, X_N\}$, which has entropy $H_{\underline{X}}$. Then*

1. *For any $N \geq 1$:*
$$H_{\underline{X}} \leq L_N^* \leq H_{\underline{X}} + 1\,. \tag{1.32}$$
2. *If the source has a finite entropy rate $h = \lim_{N\to\infty} H_{\underline{X}}/N$, then*
$$\lim_{N\to\infty} \frac{1}{N} L_N^* = h\,. \tag{1.33}$$

Proof: The basic idea in the proof of Eq. (1.32) is that, if the codewords were too short, the code wouldn't be instantaneous. **Kraft's inequality** makes this simple remark more precise. For any instantaneous code w, the lengths $l_w(\underline{x})$ satisfy:

$$\sum_{\underline{x}\in\mathcal{X}^N} 2^{-l_w(\underline{x})} \leq 1\,. \tag{1.34}$$

This fact is easily proved by representing the set of codewords as a set of leaves on a binary tree (see Fig. 1.2). Let L_M be the length of the longest codeword . Consider the set of all the 2^{L_M} possible vertices in the binary tree which are at the generation L_M, let us call them the 'descendants'. If the information $\underline{x}$ is associated with a codeword at generation l (i.e. $l_w(\underline{x}) = l$), there can be no other codewords in the branch of the tree rooted on this codeword, because the code is instantaneous. We 'erase' the corresponding $2^{L_M - l}$ descendants which cannot be codewords. The subsets of erased descendants associated with each codeword are not overlapping. Therefore the total number of erased descendants, $\sum_{\underline{x}} 2^{L_M - l_w(\underline{x})}$, must be smaller or equal to the total number of descendants, 2^{L_M}. This establishes Kraft's inequality.

Conversely, for any set of lengths $\{l(\underline{x})\}_{\underline{x}\in\mathcal{X}^N}$ which satisfies Kraft's inequality (1.34), there exists at least a code, whose codewords have the lengths $\{l(\underline{x})\}_{\underline{x}\in\mathcal{X}^N}$. A possible construction is obtained as follows. Consider the smallest length $l(\underline{x})$ and take the first allowed binary sequence of length $l(\underline{x})$ to be the codeword for $\underline{x}$. Repeat this operation with the next shortest length, and so on until you have exhausted all the codewords. It is easy to show that this procedure is successful if Eq. (1.34) is satisfied.

The problem is therefore reduced to finding the set of codeword lengths $l(\underline{x}) = l^*(\underline{x})$ which minimize the average length $L = \sum_{\underline{x}} p(\underline{x}) l(\underline{x})$ subject to Kraft's inequality (1.34). Supposing first that $l(\underline{x})$ can take arbitrary non-negative real values, this is easily done with Lagrange multipliers, and leads to $l(\underline{x}) = -\log_2 p(\underline{x})$. This set of optimal lengths, which in general cannot be realized because some of the $l(\underline{x})$ are not

integers, gives an average length equal to the entropy $H_{\underline{X}}$. It implies the lower bound in (1.32). In order to build a real code with integer lengths, we use

$$l^*(\underline{x}) = \lceil -\log_2 p(\underline{x}) \rceil \,. \tag{1.35}$$

Such a code satisfies Kraft's inequality, and its average length is less or equal than $H_{\underline{X}} + 1$, proving the upper bound in (1.32).

The second part of the theorem is a straightforward consequence of the first one. □

The code we have constructed in the proof is often called a **Shannon code**. For long strings ($N \gg 1$), it gets close to optimal. However it has no reason to be optimal in general. For instance if only one $p(x)$ is very small, it will assign to x to a very long codeword , while shorter codewords are available. It is interesting to know that, for a given source $\{X_1, \ldots, X_N\}$, there exists an explicit construction of the optimal code, called Huffman's code.

At first sight, it may appear that Theorem 1.16, together with the construction of Shannon codes, completely solves the source coding problem. Unhappily this is far from true, as the following arguments show.

From a computational point of view, the encoding procedure described above is unpractical when N is large. One can build the code once for all, and store it somewhere, but this requires $\Theta(|\mathcal{X}|^N)$ memory. On the other hand, one could reconstruct the code each time a string requires to be encoded, but this takes $\Theta(|\mathcal{X}|^N)$ operations. One can use the same code and be a bit smarter in the encoding procedure, but this does not yield a big improvement. (The symbol Θ means 'of the order of'; the precise definition is given in Appendix A.)

From a practical point of view, the construction of a Shannon code requires an accurate knowledge of the probabilistic law of the source. Suppose now you want to compress the complete works of Shakespeare. It is exceedingly difficult to construct a good model for the source 'Shakespeare'. Even worse: when you will finally have such a model, it will be of little use to compress Dante or Racine.

Happily, source coding has made tremendous progresses in both directions in the last half century. However in this book we will focus on another crucial aspect of information theory, the transmission of information.

1.6 Data transmission

We have just seen how to encode some information in a string of symbols (we used bits, but any finite alphabet is equally good). Suppose now we want to communicate this string. When the string is transmitted, it may be corrupted by some noise, which depends on the physical device used in the transmission. One can reduce this problem by adding redundancy to the string. The redundancy is to be used to correct some transmission errors, in the same way as redundancy in the English language can be used to correct some of the typos in this book. This is the domain of **channel coding**. A central result in information theory, again due to Shannon's pioneering work in 1948, relates the level of redundancy to the maximal level of noise that can be tolerated for error-free transmission. As in source coding , entropy again plays a key role in this result. This is not surprising in view of the duality between the two problems. In data

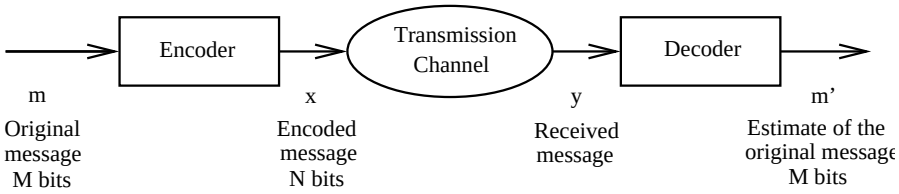

Fig. 1.3 Typical flowchart of a communication device.

compression , one wants to reduce the redundancy of the data, and the entropy gives a measure of the ultimate possible reduction. In data transmission, one wants to add some well tailored redundancy to the data.

1.6.1 Communication channels

The typical flowchart of a communication system is shown in Fig. 1.3. It applies to situations as diverse as communication between the earth and a satellite, cellular phones, or storage within the hard disk of your computer. Alice wants to send a message m to Bob. Let us assume that m is a M bit sequence. This message is first encoded into a longer one, a N bit message denoted by $\underline{x}$ with $N > M$, where the added bits will provide the redundancy used to correct for transmission errors. The encoder is a map from $\{0,1\}^M$ to $\{0,1\}^N$. The encoded message is sent through the communication channel. The output of the channel is a message $\underline{y}$. In a noiseless channel, one would simply have $\underline{y} = \underline{x}$. In a realistic channel, $\underline{y}$ is in general a string of symbols different from $\underline{x}$. Notice that $\underline{y}$ is not even necessarily a string of bits. The **channel** will be described by the transition probability $Q(\underline{y}|\underline{x})$. This is the probability that the received signal is $\underline{y}$, conditional to the transmitted signal being $\underline{x}$. Different physical channels will be described by different $Q(\underline{y}|\underline{x})$ functions. The decoder takes the message $\underline{y}$ and deduces from it an estimate m' of the sent message.

Exercise 1.7 Consider the following example of a channel with **insertions**. When a bit `x` is fed into the channel, either `x` or `x0` are received with equal probability 1/2. Suppose that you send the string `111110`. The string `1111100` will be received with probability $2 \cdot 1/64$ (the same output can be produced by an error either on the 5^{th} or on the 6^{th} digit). Notice that the output of this channel is a bit string which is always longer or equal to the transmitted one.

A simple code for this channel is easily constructed: use the string `100` for each `0` in the original message and `1100` for each `1`. Then for instance you have the encoding

$$\texttt{01101} \mapsto \texttt{100110011001001100}\,. \tag{1.36}$$

The reader is invited to define a decoding algorithm and verify its effectiveness.

Hereafter we shall consider **memoryless** channels. In this case, for any input $\underline{x} = (x_1, ..., x_N)$, the output message is a string of N letters, $\underline{y} = (y_1, ..., y_N)$, from an alphabet $\mathcal{Y} \ni y_i$ (not necessarily binary). In memoryless channels, the noise acts

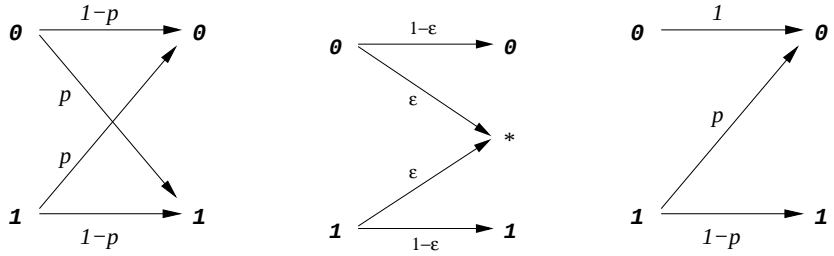

Fig. 1.4 Three communication channels. *Left*: the binary symmetric channel. An error in the transmission, in which the output bit is the opposite of the input one, occurs with probability p. *Middle*: the binary erasure channel. An error in the transmission, signaled by the output $*$, occurs with probability ϵ. *Right*: the Z channel. An error occurs with probability p whenever a 1 is transmitted.

independently on each bit of the input. This means that the conditional probability $Q(\underline{y}|\underline{x})$ factorizes:

$$Q(\underline{y}|\underline{x}) = \prod_{i=1}^{N} Q(y_i|x_i) \ , \tag{1.37}$$

and the transition probability $Q(y_i|x_i)$ is i independent.

Example 1.17 Binary symmetric channel (BSC). The input x_i and the output y_i are both in $\{0, 1\}$. The channel is characterized by one number, the probability p that the channel output is different from the input, also called the **crossover** (or **flip**) probability. It is customary to represent it by the diagram of Fig. 1.4.

Example 1.18 Binary erasure channel (BEC). In this case some of the input bits are erased instead of being corrupted: x_i is still in $\{0, 1\}$, but y_i now belongs to $\{0, 1, *\}$, where $*$ means that the symbol has been erased. In the symmetric case, this channel is described by a single number, the probability ϵ that a bit is erased, see Fig. 1.4.

Example 1.19 Z channel. In this case the output alphabet is again $\{0, 1\}$. Moreover, a 0 is always transmitted correctly, while a 1 becomes a 0 with probability p. The name of this channel come from its graphical representation, see Fig. 1.4.

A very important characteristic of a channel is the **channel capacity** C. It is defined in terms of the mutual information $I_{X,Y}$ of the variables X (the bit which was sent) and Y (the signal which was received), through:

$$C = \max_{p(x)} I_{X,Y} = \max_{p(x)} \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} p(x, y) \log_2 \frac{p(x, y)}{p(x)p(y)} \tag{1.38}$$

We recall that in our case $p(x,y) = p(x)Q(y|x)$, and $I_{X,Y}$ measures the reduction on the uncertainty of x due to the knowledge of y. The capacity C gives a measure of how faithful a channel can be: If the output of the channel is pure noise, x and y are uncorrelated and $C = 0$. At the other extreme if $y = f(x)$ is known for sure, given x, then $C = \max_{\{p(x)\}} H(p) = 1$ bit (for binary inputs). The interest of the capacity will become clear in section 1.6.3 with Shannon's coding theorem which shows that C characterizes the amount of information which can be transmitted faithfully through a channel.

Example 1.20 Consider a binary symmetric channel with flip probability p. Let us call q the probability that the source sends $x = \mathtt{0}$, and $1-q$ the probability of $x = \mathtt{1}$. It is easy to show that the mutual information in Eq. (1.38) is maximized when zeros and ones are transmitted with equal probability (i.e. when $q = 1/2$).

Using the expression (1.38), we get, $C = 1 - \mathcal{H}(p)$ bits, where $\mathcal{H}(p)$ is the entropy of Bernoulli's process with parameter p (plotted in Fig. 1.1).

Example 1.21 Consider now the binary erasure channel with error probability ϵ. The same argument as above applies. It is therefore easy to get $C = 1 - \epsilon$.

Exercise 1.8 Compute the capacity of the Z channel.

1.6.2 Error correcting codes

We need one last ingredient in order to have a complete definition of the channel coding problem: the behavior of the information source. We shall assume that the source produces a sequence of uncorrelated unbiased bits. This may seem at first a very crude model for any real information source. Surprisingly, Shannon's source-channel separation theorem assures that there is indeed no loss of generality in treating this case.

The sequence of bits produced by the source is divided into blocks $m_1, m_2, m_3, \dots$ of length M. The **encoding** is a mapping from $\{0,1\}^M \ni m$ to $\{0,1\}^N$, with $N \geq M$. Each possible M-bit message m is mapped to a **codeword** $\underline{x}(m)$ which can be seen as a point in the N-dimensional unit hypercube. The codeword length N is also called the **blocklength**. There are 2^M codewords, and the set of all possible codewords is called the **codebook**. When the message is transmitted, the codeword $\underline{x}$ is corrupted to $\underline{y} \in \mathcal{Y}^N$ with probability $Q(\underline{y}|\underline{x}) = \prod_{i=1}^N Q(y_i|x_i)$. The output alphabet $\mathcal{Y}$ depends on the channel. The **decoder** is a mapping from $\mathcal{Y}^N$ to $\{0,1\}^M$ which takes the received message $\underline{y} \in \mathcal{Y}^N$ and maps it to one of the possible original messages $m' = d(\underline{y}) \in \{0,1\}^M$.

An **error correcting code** is defined by the set of two functions, the encoding $\underline{x}(m)$ and the decoding $d(\underline{y})$. The ratio

$$R = \frac{M}{N} \tag{1.39}$$

of the original number of bits to the transmitted number of bits is called the **rate** of the code. The rate is a measure of the redundancy of the code. The smaller the rate, the more redundancy is added to the code, and the more errors one should be able to correct.

The **block error probability** of a code on the input message m, denoted by $\mathrm{P_B}(m)$, is the probability that the decoded message differs from the one which was sent:

$$\mathrm{P_B}(m) = \sum_{\underline{y}} Q(\underline{y}|\underline{x}(m)) \;\; \mathbb{I}(d(\underline{y}) \neq m) \; . \tag{1.40}$$

Knowing the error probability for each possible transmitted message is an exceedingly detailed characterization of the code performances. One can therefore introduce a **maximal block error probability** as

$$\mathrm{P_B^{max}} \equiv \max_{m \in \{0,1\}^M} \mathrm{P_B}(m) \, . \tag{1.41}$$

This corresponds to characterizing the code by its 'worst case' performances. A more optimistic point of view consists in averaging over the input messages. Since we assumed all of them to be equiprobable, we introduce the **average block error probability** as

$$\mathrm{P_B^{av}} \equiv \frac{1}{2^M} \sum_{m \in \{0,1\}^M} \mathrm{P_B}(m) \, . \tag{1.42}$$

Since this is a very common figure of merit for error correcting codes, we shall call it block error probability and use the symbol $\mathrm{P_B}$ without further specification hereafter.

Example 1.22 Repetition code. Consider a BSC which transmits a wrong bit with probability p. A simple code consists in repeating k times each bit, with k odd. Formally we have $M = 1$, $N = k$ and

$$\underline{x}(\mathtt{0}) = \underbrace{\mathtt{000\ldots00}}_{k} \, , \tag{1.43}$$

$$\underline{x}(\mathtt{1}) = \underbrace{\mathtt{111\ldots11}}_{k} \, . \tag{1.44}$$

This code has rate $R = M/N = 1/k$. For instance with $k = 3$, the original stream `0110001` is encoded as `000111111000000000111`. A possible decoder consists in parsing the received sequence in groups of k bits, and finding the message m' using a majority rule among the k bits. In our example with $k = 3$, if the received group of three bits is `111` or `110` or any permutation, the corresponding input bit is assigned to `1`, otherwise it is assigned to `0`. For instance if the channel output is `000101111011000010111`, this decoder returns `0111001`.

Exercise 1.9 The k-repetition code corrects up to $\lfloor k/2 \rfloor$ errors per group of k bits. Show that the block error probability for general k is

$$\mathrm{P_B} = \sum_{r=\lceil k/2 \rceil}^{k} \binom{k}{r} (1-p)^{k-r} p^r \ . \tag{1.45}$$

Notice that, for any finite k and $p > 0$, $\mathrm{P_B}$ is strictly positive. In order to have $\mathrm{P_B} \to 0$ we must consider $k \to \infty$. Since the rate is $R = 1/k$, the price to pay for a vanishing block error probability is a vanishing communication rate!

Happily enough, we will see that much better codes exist.

1.6.3 The channel coding theorem

Consider a communication channel whose capacity (1.38) is C. In his seminal 1948 paper, Shannon proved the following theorem.

Theorem 1.23 *For every rate $R < C$, there exists a sequence of codes $\{\mathcal{C}_N\}$, of blocklength N, rate R_N, and block error probability $\mathrm{P_{B},}_N$, such that $R_N \to R$ and $\mathrm{P_{B},}_N \to 0$ as $N \to \infty$. Conversely, if for a sequence of codes $\{\mathcal{C}_N\}$, one has $R_N \to R$ and $\mathrm{P_{B},}_N \to 0$ as $N \to \infty$, then $R < C$.*

In practice, for long messages (i.e. large N), reliable communication is possible if and only if the communication rate stays below capacity. The direct part of the proof will be given in Sec. 6.4 using the random code ensemble. We shall not give a full proof of the converse part in general, but only in the case of the BSC channel, in Sec. 6.5.2. Here we keep to some qualitative comments and provide the intuitive idea underlying this theorem.

First of all, the result is rather surprising when one meets it for the first time. As we saw on the example of repetition codes above, simple minded codes typically have a positive error probability, for any non-vanishing noise level. Shannon's theorem establishes that it is possible to achieve vanishing error probability, while keeping the communication rate bounded away from zero.

One can get an intuitive understanding of the role of the capacity through a qualitative reasoning, which uses the fact that a random variable with entropy H 'typically' takes 2^H values. For a given codeword $\underline{x}(m) \in \{0,1\}^N$, the channel output $\underline{y}$ is a random variable with an entropy $H_{\underline{y}|\underline{x}} = NH_{y|x}$. There exist about $2^{NH_{y|x}}$ such outputs. For a perfect decoding, one needs a decoding function $d(\underline{y})$ that maps each of them to the original message m. Globally, the typical number of possible outputs is 2^{NH_y}, therefore one can distinguish at most $2^{N(H_y - H_{y|x})}$ codewords. In order to have vanishing maximal error probability, one needs to be able to send all the $2^M = 2^{NR}$ codewords. This is possible only if $R < H_y - H_{y|x} \leq C$.

Notes

There are many textbooks introducing to probability and to information theory. A classic probability textbook is (Feller, 1968). For a more recent reference see (Durrett, 1995). The original Shannon paper (Shannon, 1948) is universally recognized as the foundation of information theory. A very nice modern introduction to the subject is the book (Cover and Thomas, 1991). The reader may find there a description of Huffman codes which we did not treat in the present Chapter, as well as more advanced topics in source coding .

We did not show that the six properties listed in Sec. 1.2 provide in fact an alternative (axiomatic) definition of entropy. The interested reader is referred to (Csiszár and Körner, 1981). An advanced information theory book with much space devoted to coding theory is (Gallager, 1968). The recent and very rich book (MacKay, 2002) discusses the relations with statistical inference and machine learning.

The information-theoretic definition of entropy has been used in many contexts. It can be taken as a founding concept in statistical mechanics. This approach, pioneered in (Jaynes, 1957), is discussed in (Balian, 1992).

2

Statistical physics and probability theory

One of the greatest achievement of science has been to realize that matter is made out of a small number of simple elementary components. This result seems to be in striking contrast with our experience. Both at a simply perceptual level and with more refined scientific experience, we come in touch with an ever-growing variety of states of the matter with disparate properties. The ambitious purpose of statistical physics (and, more generally, of a large branch of condensed matter physics) is to understand this variety. It aims at explaining how complex behaviors can emerge when large numbers of identical elementary components are allowed to interact.

We have, for instance, experience of water in three different states (solid, liquid and gaseous). Water molecules and their interactions do not change when passing from one state to the other. Understanding how the same interactions can result in qualitatively different macroscopic states, and what rules the change of state, is a central topic of statistical physics.

The foundations of statistical physics rely on two important steps. The first one consists in passing form the deterministic laws of physics, like Newton's law, to a probabilistic description. The idea is that a precise knowledge of the motion of each molecule in a macroscopic system is inessential to the understanding of the system as a whole: instead, one can postulate that the microscopic dynamics, because of its chaoticity, allows for a purely probabilistic description. The detailed justification of this basic step has been achieved only in a small number of concrete cases. Here we shall bypass any attempt at such a justification: we directly adopt a purely probabilistic point of view, as a basic postulate of statistical physics.

The second step starts from the probabilistic description and recovers determinism at a macroscopic level by some sort of law of large numbers. We all know that water boils at 100° Celsius (at atmospheric pressure) or that its density (at 25° Celsius and atmospheric pressures) is $1\,\mathrm{gr/cm^3}$. The regularity of these phenomena is not related to the deterministic laws which rule the motions of water molecule. It is instead the consequence of the fact that, because of the large number of particles involved in any macroscopic system, fluctuations are "averaged out". We shall discuss this kind of phenomena in Sec. 2.4 and, more mathematically, in Ch. **??**.

The purpose of this Chapter is to introduce the most basic concepts of this discipline, for an audience of non-physicists with a mathematical background. We adopt a somewhat restrictive point of view, which keeps to classical (as opposed to quantum) statistical physics, and basically describes it as a branch of probability theory (Secs.

2.1 to 2.3). In Section 2.4 we focus on large systems, and stress that the statistical physics approach becomes particularly meaningful in this regime. Theoretical statistical physics often deals with highly idealized mathematical models of real materials. The most interesting (and challenging) task is in fact to understand the *qualitative* behavior of such systems. With this aim, one can discard any "irrelevant" microscopic detail from the mathematical description of the model. In Sec. 2.5, the study of ferromagnetism through the introduction of the Ising model gives an example of this modelization procedure. Compared to the case of Ising ferromagnets, the theoretical understanding of spin glasses is much less developed. Section 2.6 presents a rapid preview of this fascinating subject.

2.1 The Boltzmann distribution

The basic ingredients for a probabilistic description of a physical system are:

- A **space of configurations** $\mathcal{X}$. One should think of $x \in \mathcal{X}$ as giving a complete microscopic determination of the state of the system under consideration. We are not interested in defining the most general mathematical structure for $\mathcal{X}$ such that a statistical physics formalism can be constructed. Throughout this book we will in fact consider only two very simple types of configuration spaces: (i) finite sets, and (ii) smooth, compact, finite-dimensional manifolds. If the system contains N 'particles', the configuration space is a product space:
$$\mathcal{X}_N = \underbrace{\mathcal{X} \times \cdots \times \mathcal{X}}_{N} . \tag{2.1}$$
The configuration of the system has the form $\underline{x} = (x_1, \ldots, x_N)$. Each coordinate $x_i \in \mathcal{X}$ is meant to represent the state (position, orientation, etc) of one of the particles.
But for a few examples, we shall focus on configuration spaces of type (i). We will therefore adopt a discrete-space notation for $\mathcal{X}$. The generalization to continuous configuration spaces is in most cases intuitively clear (although it may present some technical difficulties).
- A set of **observables**, which are real-valued functions on the configuration space $\mathcal{O} : x \mapsto \mathcal{O}(x)$. If $\mathcal{X}$ is a manifold, we shall limit ourselves to observables which are smooth functions of the configuration x. Observables are physical quantities which can be measured through an experiment (at least in principle).
- Among all the observables, a special role is played by the **energy function** $E(x)$. When the system is a N particle system, the energy function generally takes the form of sums of terms involving few particles. An energy function of the form:
$$E(\underline{x}) = \sum_{i=1}^{N} E_i(x_i) \tag{2.2}$$
corresponds to a **non-interacting** system. An energy of the form
$$E(\underline{x}) = \sum_{i_1,..,i_k} E_{i_1,..,i_k}(x_{i_1}, ..., x_{i_k}) \tag{2.3}$$

is called a k**-body** interaction. In general, the energy will contain some pieces involving k-body interactions, with $k \in \{1, 2, ..., K\}$. An important feature of real physical systems is that K is never a large number (usually $K = 2$ or 3), even when the number of particles N is very large. The same property holds for all measurable observables. However, for the general mathematical formulation which we will use here, the energy can be any real valued function on $\mathcal{X}$.

Once the configuration space $\mathcal{X}$ and the energy function are fixed, the probability $\mu_\beta(x)$ for the system to be found in the configuration x is given by the **Boltzmann's distribution**:

$$\mu_\beta(x) = \frac{1}{Z(\beta)}\, e^{-\beta E(x)}\,, \qquad Z(\beta) = \sum_{x\in\mathcal{X}} e^{-\beta E(x)}\,. \tag{2.4}$$

The real parameter $T = 1/\beta$ is the **temperature** (and one refers to β as the inverse temperature). Note that the temperature is usually defined as $T = 1/(k_{\rm B}\beta)$ where the value of $k_{\rm B}$, Boltzmann's constant, depends on the unit of measure for temperature. Here we adopt the simple choice $k_{\rm B} = 1$. The normalization constant $Z(\beta)$ is called the **partition function**. Notice that Eq. (2.4) defines indeed the density of the Boltzmann distribution with respect to some reference measure. The reference measure is usually the counting measure if $\mathcal{X}$ is discrete or the Lebesgue measure if $\mathcal{X}$ is continuous. It is customary to denote the expectation value with respect to Boltzmann's measure by brackets: the expectation value $\langle \mathcal{O}(x) \rangle$ of an observable $\mathcal{O}(x)$, also called its **Boltzmann average** is given by:

$$\langle \mathcal{O} \rangle = \sum_{x\in\mathcal{X}} \mu_\beta(x)\mathcal{O}(x) = \frac{1}{Z(\beta)} \sum_{x\in\mathcal{X}} e^{-\beta E(x)}\mathcal{O}(x)\,. \tag{2.5}$$

Example 2.1 One intrinsic property of elementary particles is their spin. For 'spin 1/2' particles, the spin σ takes only two values: $\sigma = \pm 1$. A localized spin 1/2 particle, whose only degree of freedom is the spin, is described by $\mathcal{X} = \{+1, -1\}$, and is called an **Ising spin**. The energy of the spin in the state $\sigma \in \mathcal{X}$ in a magnetic field B is

$$E(\sigma) = -B\,\sigma \tag{2.6}$$

Boltzmann's probability of finding the spin in the state σ is

$$\mu_\beta(\sigma) = \frac{1}{Z(\beta)}\, e^{-\beta E(\sigma)} \qquad Z(\beta) = e^{-\beta B} + e^{\beta B} = 2\cosh(\beta B)\,. \tag{2.7}$$

The average value of the spin, called the **magnetization** is

$$\langle \sigma \rangle = \sum_{\sigma \in \{1,-1\}} \mu_\beta(\sigma)\, \sigma = \tanh(\beta B)\,. \tag{2.8}$$

At high temperatures, $T \gg |B|$, the magnetization is small. At low temperatures, the magnetization its close to its maximal value, $\langle \sigma \rangle = 1$ if $B > 0$. Section 2.5 will discuss the behaviors of many Ising spins, with some more complicated energy functions.

Example 2.2 Some spin variables can have a larger space of possible values. For instance a **Potts spin** with q states takes values in $\mathcal{X} = \{1, 2, \ldots, q\}$. In presence of a magnetic field of intensity h pointing in direction $r \in \{1, \ldots, q\}$, the energy of the Potts spin is

$$E(\sigma) = -B\,\mathbb{I}(\sigma = r)\ . \tag{2.9}$$

In this case, the average value of the spin in the direction of the field is

$$\langle \mathbb{I}(\sigma = r) \rangle = \frac{\exp(\beta B)}{\exp(\beta B) + (q-1)}\ . \tag{2.10}$$

Example 2.3 Let us consider a single water molecule inside a closed container, for instance, inside a bottle. A water molecule H_2O is already a complicated object. In a first approximation, we can neglect its structure and model the molecule as a point inside the bottle. The space of configurations reduces then to:

$$\mathcal{X} = \texttt{BOTTLE} \subset \mathbb{R}^3 \,, \tag{2.11}$$

where we denoted by BOTTLE the region of $\mathbb{R}^3$ delimited by the container. Notice that this description is not very accurate at a microscopic level.

The description of the precise form of the bottle can be quite complex. On the other hand, it is a good approximation to assume that all positions of the molecule are equiprobable: the energy is independent of the particle's position $x \in \texttt{BOTTLE}$. One has then:

$$\mu(x) = \frac{1}{Z} \,, \qquad Z = |\mathcal{X}| \,, \tag{2.12}$$

and the Boltzmann average of the particle's position, $\langle x \rangle$, is the barycenter of the bottle.

Example 2.4 In assuming that all the configurations of the previous example are equiprobable, we neglected the effect of gravity on the water molecule. In the presence of gravity our water molecule at position x has an energy:

$$E(x) = w\, \mathrm{h}(x) \,, \tag{2.13}$$

where $\mathrm{h}(x)$ is the height corresponding to the position x and w is a positive constant, determined by terrestrial attraction, which is proportional to the mass of the molecule. Given two positions x and y in the bottle, the ratio of the probabilities to find the particle at these positions is

$$\frac{\mu_\beta(x)}{\mu_\beta(y)} = \exp\{-\beta w[\mathrm{h}(x) - \mathrm{h}(y)]\} \tag{2.14}$$

For a water molecule at a room temperature of 20 degrees Celsius ($T = 293$ degrees Kelvin), one has $\beta w \approx 7 \times 10^{-5}\,\mathrm{m}^{-1}$. Given a point x at the bottom of the bottle and y at a height of 20 cm, the probability to find a water molecule 'near' x is approximately 1.000014 times larger than the probability to find it 'near' y. For a tobacco-mosaic virus, which is about 2×10^6 times heavier than a water molecule, the ratio is $\mu_\beta(x)/\mu_\beta(y) \approx 1.4 \times 10^{12}$ which is very large. For a grain of sand the ratio is so large that one never observes it floating around y. Notice that, while these ratios of probability densities are easy to compute, the partition function and therefore the absolute values of the probability densities can be much more complicated to estimate, depending on the shape of the bottle.

Example 2.5 In many important cases, we are given the space of configurations $\mathcal{X}$ and a stochastic dynamics defined on it. The most interesting probability distribution for such a system is the stationary state $\mu_{\rm st}(x)$ (we assume that it is unique). For sake of simplicity, we can consider a finite space $\mathcal{X}$ and a discrete time Markov chain with transition probabilities $\{w(x \to y)\}$ (in Chapter **??** we shall recall some basic definitions concerning Markov chains). It happens sometimes that the transition rates satisfy, for any couple of configurations $x, y \in \mathcal{X}$, the relation

$$f(x)w(x \to y) = f(y)w(y \to x)\,, \tag{2.15}$$

for some positive function $f(x)$. As we shall see in Chapter **??**, when this condition, called **detailed balance**, is satisfied (together with a couple of other technical conditions), the stationary state has the Boltzmann form (2.4) with $e^{-\beta E(x)} = f(x)$.

Exercise 2.1 As a particular realization of the above example, consider an 8×8 chessboard and a special piece sitting on it. At any time step the piece will stay still (with probability $1/2$) or move randomly to one of the neighboring positions (with probability $1/2$). Does this process satisfy the condition (2.15)? Which positions on the chessboard have lower (higher) "energy"? Compute the partition function.

From a purely probabilistic point of view, one can wonder why one bothers to decompose the distribution $\mu_\beta(x)$ into the two factors $e^{-\beta E(x)}$ and $1/Z(\beta)$. Of course the motivations for writing the Boltzmann factor $e^{-\beta E(x)}$ in exponential form come essentially from physics, where one knows (either exactly or within some level of approximation) the form of the energy. This also justifies the use of the inverse temperature β (after all, one could always redefine the energy function in such a way to set $\beta = 1$).

However, even if we adopt a mathematical viewpoint, and if we are interested in a particular distribution $\mu(x)$ which corresponds to a particular value of the temperature, it is often illuminating to embed it into a one-parameter family as is done in the Boltzmann expression (2.4). Indeed, (2.4) interpolates smoothly between several interesting situations. As $\beta \to 0$ (**high-temperature limit**), one recovers the uniform probability distribution

$$\lim_{\beta \to 0} \mu_\beta(x) = \frac{1}{|\mathcal{X}|}\,. \tag{2.16}$$

Both the probabilities $\mu_\beta(x)$ and the observables expectation values $\langle \mathcal{O}(x) \rangle$ can be expressed as convergent Taylor expansions around $\beta = 0$. For small β the Boltzmann distribution can be seen as a "softening" of the original one.

In the limit $\beta \to \infty$ (**low-temperature limit**), the Boltzmann distribution concentrates on the global maxima of the original one. More precisely, a configuration $x_0 \in \mathcal{X}$ such that $E(x) \geq E(x_0)$ for any $x \in \mathcal{X}$ is called a **ground state**. The minimum value of the energy $E_0 = E(x_0)$ is called the **ground state energy**. We will

denote the set of ground states as $\mathcal{X}_0$. It is elementary to show that, for a discrete configuration space:

$$\lim_{\beta\to\infty} \mu_\beta(x) = \frac{1}{|\mathcal{X}_0|}\, \mathbb{I}(x \in \mathcal{X}_0)\,, \tag{2.17}$$

where $\mathbb{I}(x \in \mathcal{X}_0) = 1$ if $x \in \mathcal{X}_0$ and $\mathbb{I}(x \in \mathcal{X}_0) = 0$ otherwise. The above behavior is summarized in physicists jargon by saying that, at low temperature, "low energy configurations dominate" the behavior of the system.

2.2 Thermodynamic potentials

Several properties of the Boltzmann distribution (2.4) are conveniently summarized through the thermodynamic potentials. These are functions of the temperature $1/\beta$ and of the various parameters defining the energy $E(x)$. The most important thermodynamic potential is the **free-energy**:

$$F(\beta) = -\frac{1}{\beta}\, \log Z(\beta)\,, \tag{2.18}$$

where $Z(\beta)$ is the partition function already defined in Eq. (2.4). The factor $-1/\beta$ in Eq. (2.18) is due essentially to historical reasons. In calculations it is often more convenient to use the **free-entropy**[1] $\Phi(\beta) = -\beta F(\beta) = \log Z(\beta)$.

Two more thermodynamic potentials are derived from the free-energy: the **internal energy** $U(\beta)$ and the **canonical entropy** $S(\beta)$:

$$U(\beta) = \frac{\partial}{\partial\beta}(\beta F(\beta))\,, \qquad S(\beta) = \beta^2 \frac{\partial F(\beta)}{\partial\beta}\,. \tag{2.19}$$

By direct computation one obtains the following identities concerning the potentials defined so far:

$$F(\beta) = U(\beta) - \frac{1}{\beta}\, S(\beta) = -\frac{1}{\beta}\Phi(\beta)\,, \tag{2.20}$$

$$U(\beta) = \langle E(x)\rangle\,, \tag{2.21}$$

$$S(\beta) = -\sum_x \mu_\beta(x) \log \mu_\beta(x)\,, \tag{2.22}$$

$$-\frac{\partial^2}{\partial\beta^2}(\beta F(\beta)) = \langle E(x)^2\rangle - \langle E(x)\rangle^2\,. \tag{2.23}$$

For discrete $\mathcal{X}$, equation (2.22) can be rephrased by saying that the canonical entropy is the Shannon entropy of the Boltzmann distribution, as we defined it in Ch. 1. It implies that $S(\beta) \geq 0$. Equation (2.23) implies that the free-entropy is a convex function of the temperature. Finally, Eq. (2.21) justifies the name "internal energy" for $U(\beta)$.

[1] Unlike the other potentials, there is no universally accepted name for $\Phi(\beta)$; because this potential is very useful, we adopt for it the name 'free-entropy'

In order to have some intuition of the content of these definitions, let us reconsider the high- and low-temperature limits already treated in the previous Section. In the high-temperature limit, $\beta \to 0$, one finds

$$F(\beta) = -\frac{1}{\beta} \log |\mathcal{X}| + \langle E(x) \rangle_0 + \Theta(\beta) \,, \tag{2.24}$$

$$U(\beta) = \langle E(x) \rangle_0 + \Theta(\beta) \,, \tag{2.25}$$

$$S(\beta) = \log |\mathcal{X}| + \Theta(\beta) \,. \tag{2.26}$$

(Recall that Θ stands for 'of the order of,' cf. Appendix A). The interpretation of these formulae is straightforward. At high temperature the system can be found in any possible configuration with similar probabilities (the probabilities being exactly equal when $\beta = 0$). The entropy counts the number of possible configurations. The internal energy is just the average value of the energy over the configurations with uniform probability.

While the high temperature expansions (2.24)–(2.26) have the same form both for a discrete and a continuous configuration space $\mathcal{X}$, in the low temperature case, we must be more careful. If $\mathcal{X}$ is finite we can meaningfully define the **energy gap** $\Delta E > 0$ as follows (recall that we denoted by E_0 the ground-state energy)

$$\Delta E = \min\{E(y) - E_0 \,:\, y \in \mathcal{X} \backslash \mathcal{X}_0\} \,. \tag{2.27}$$

With this definition we get

$$F(\beta) = E_0 - \frac{1}{\beta} \log |\mathcal{X}_0| + \Theta(e^{-\beta \Delta E}) \,, \tag{2.28}$$

$$E(\beta) = E_0 + \Theta(e^{-\beta \Delta E}) \,, \tag{2.29}$$

$$S(\beta) = \log |\mathcal{X}_0| + \Theta(e^{-\beta \Delta E}) \,. \tag{2.30}$$

The interpretation is that, at low temperature, the system is found with equal probability in any of the ground states, and nowhere else. Once again the entropy counts the number of available configurations and the internal energy is the average of their energies (which coincide with the ground state).

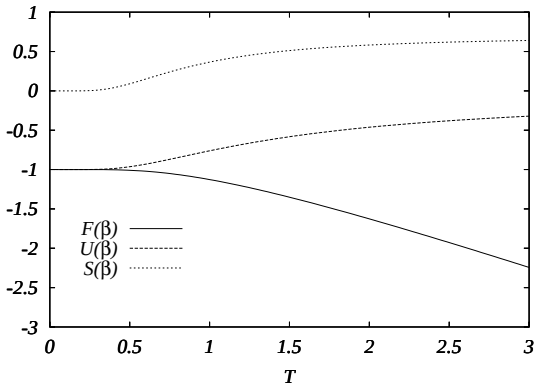

Fig. 2.1 Thermodynamic potentials for a two-level system with $\epsilon_1 = -1$, $\epsilon_2 = +1$ as a function of the temperature $T = 1/\beta$.

Exercise 2.2 A two level system. This is the simplest non-trivial example: $\mathcal{X} = \{1, 2\}$, $E(1) = \epsilon_1$, $E(2) = \epsilon_2$. Without loss of generality we assume $\epsilon_1 < \epsilon_2$. It can be used as a mathematical model for many physical systems, like the spin 1/2 particle discussed above.

Derive the following results for the thermodynamic potentials ($\Delta = \epsilon_2 - \epsilon_1$ is the energy gap):

$$F(\beta) = \epsilon_1 - \frac{1}{\beta} \log(1 + e^{-\beta\Delta}) \,, \tag{2.31}$$

$$U(\beta) = \epsilon_1 + \frac{e^{-\beta\Delta}}{1 + e^{-\beta\Delta}} \Delta \,, \tag{2.32}$$

$$S(\beta) = \frac{e^{-\beta\Delta}}{1 + e^{-\beta\Delta}} \beta\Delta + \log(1 + e^{-\beta\Delta}) \,. \tag{2.33}$$

The behavior of these functions is presented in Fig. 2.1. The reader can work out the asymptotics, and check the general high and low temperature behaviors given above.

Exercise 2.3 We come back to the example of the previous section: one water molecule, modeled as a point, in a bottle. Moreover, we consider the case of a cylindrical bottle of base $B \subset \mathbb{R}^2$ (surface $|B|$) and height d.

Using the energy function (2.13), derive the following explicit expressions for the thermodynamic potentials:

$$F(\beta) = -\frac{1}{\beta} \log |B| - \frac{1}{\beta} \log \frac{1 - e^{-\beta w d}}{\beta w} \,, \tag{2.34}$$

$$U(\beta) = \frac{1}{\beta} - \frac{wd}{e^{\beta w d} - 1} \,, \tag{2.35}$$

$$S(\beta) = \log |Bd| + 1 - \frac{\beta w d}{e^{\beta w d} - 1} - \log \left(\frac{\beta w d}{1 - e^{-\beta w d}} \right) . \tag{2.36}$$

Notice that the internal energy formula can be used to compute the average height of the molecule $\langle \mathrm{h}(x) \rangle = U(\beta)/w$. This is a consequence of the definition of the energy, cf. Eq. (2.13) and of Eq. (2.21). Plugging in the correct w constant, one may find that the average height descends below 49.99% of the bottle height $d = 20\,\mathrm{cm}$ only when the temperature is below $3.2^{\mathrm{o}}\,K$.

Exercise 2.4 Using the expressions (2.34)–(2.36), derive the low-temperature expansions:

$$F(\beta) = -\frac{1}{\beta} \log \left(\frac{|B|}{\beta w} \right) + \Theta(e^{-\beta w d}) \,, \tag{2.37}$$

$$U(\beta) = \frac{1}{\beta} + \Theta(e^{-\beta w d}) \,, \tag{2.38}$$

$$S(\beta) = \log \left(\frac{|B| e}{\beta w} \right) + \Theta(e^{-\beta w d}) \,. \tag{2.39}$$

In this case $\mathcal{X}$ is continuous, and the energy has no gap. Nevertheless these results can be understood as follows: at low temperature the molecule is confined to a layer of height of order $1/(\beta w)$ above the bottom of the bottle. It occupies therefore a volume of size $|B|/(\beta w)$. Its entropy is approximately given by the logarithm of such a volume.

Exercise 2.5 Let us reconsider the above example and assume the bottle to have a different shape, for instance a sphere of radius R. In this case it is difficult to compute explicit expressions for the thermodynamic potentials but one can easily compute the low-temperature expansions. For the entropy one gets at large β:

$$S(\beta) = \log\left(\frac{2\pi e^2 R}{\beta^2 w^2}\right) + \Theta(1/\beta)\,. \tag{2.40}$$

The reader should try understand the difference between this result and Eq. (2.39) and provide an intuitive explanation as in the previous example. Physicists say that the low-temperature thermodynamic potentials reveal the "low-energy structure" of the system.

2.3 The fluctuation dissipation relations

It often happens that the energy function depends smoothly upon some real parameters. They can be related to the experimental conditions under which a physical system is studied, or to some fundamental physical quantity. For instance, the energy of a water molecule in the gravitational field, cf. Eq. (2.13), depends upon the weight w of the molecule itself. Although this is a constant number in the physical world, it is useful, in the theoretical treatment, to consider it as an adjustable parameter.

It is therefore interesting to consider an energy function $E_\lambda(x)$ which depends smoothly upon some parameter λ and admits the following Taylor expansion in the neighborhood of $\lambda = \lambda_0$:

$$E_\lambda(x) = E_{\lambda_0}(x) + (\lambda - \lambda_0)\left.\frac{\partial E}{\partial \lambda}\right|_{\lambda_0}(x) + O((\lambda-\lambda_0)^2)\,. \tag{2.41}$$

The dependence of the free-energy and of other thermodynamic potentials upon λ in the neighborhood of λ_0 is easily related to the explicit dependence of the energy function itself. Let us consider the partition function, and expand it to first order in $\lambda - \lambda_0$:

$$\begin{aligned} Z(\lambda) &= \sum_x \exp\left(-\beta\left[E_{\lambda_0}(x) + (\lambda-\lambda_0)\left.\frac{\partial E}{\partial\lambda}\right|_{\lambda_0}(x) + O((\lambda-\lambda_0)^2)\right]\right) \\ &= Z(\lambda_0)\left[1 - \beta(\lambda-\lambda_0)\langle\left.\frac{\partial E}{\partial\lambda}\right|_{\lambda_0}\rangle_0 + O((\lambda-\lambda_0)^2)\right] \end{aligned} \tag{2.42}$$

where we denoted by $\langle\cdot\rangle_0$ the expectation with respect to the Boltzmann distribution at $\lambda = \lambda_0$.

This shows that the free-entropy behaves as:

$$\left.\frac{\partial\Phi}{\partial\lambda}\right|_{\lambda_0} = -\beta\,\langle\left.\frac{\partial E}{\partial\lambda}\right|_{\lambda_0}\rangle_0\,, \tag{2.43}$$

One can also consider the λ dependence of the expectation value of a generic observable $A(x)$. Using again the Taylor expansion one finds that

$$\left.\frac{\partial \langle A\rangle_\lambda}{\partial\lambda}\right|_{\lambda_0} = -\beta \left\langle A\,;\, \left.\frac{\partial E}{\partial\lambda}\right|_{\lambda_0} \right\rangle_0 . \tag{2.44}$$

where we denoted by $\langle A;B\rangle$ the **connected correlation function**: $\langle A;B\rangle = \langle AB\rangle - \langle A\rangle\langle B\rangle$. A particular example of this relation was given in Eq. (2.23).

The result (2.44) has important practical consequences and many generalizations. Imagine you have an experimental apparatus that allows you to tune some parameter λ (for instance the pressure of a gas, or the magnetic or electric field acting on some material) and to monitor the value of the observable $A(x)$ (the volume of the gas, the polarization or magnetization of the material). The quantity on the left-hand side of Eq. (2.44) is the response of the system to an infinitesimal variation of the tunable parameter. On the right-hand side, we find some correlation function within the "unperturbed" system. One possible application is to measure correlations within a system by monitoring its response to an external perturbation. The relation (2.44) between a correlation and a response is called **fluctuation dissipation theorem**.

2.4 The thermodynamic limit

The main purpose of statistical physics is to understand the macroscopic behavior of a large number, $N \gg 1$, of simple components (atoms, molecules, etc) when they are brought together.

To be concrete, let us consider a few drops of water in a bottle. A configuration of the system is given by the positions and orientations of all the H_2O molecules inside the bottle. In this case $\mathcal{X}$ is the set of positions and orientations of a single molecule, and N is typically of order 10^{23} (more precisely, 18 gr of water contain approximately $6\cdot 10^{23}$ molecules). The sheer magnitude of such a number leads physicists to focus on the $N\to\infty$ limit, also called the **thermodynamic limit**.

As shown by the examples below, for large N the thermodynamic potentials are often proportional to N. One is thus lead to introduce the **intensive thermodynamic potentials** as follows. Let us denote by $F_N(\beta)$, $U_N(\beta)$, $S_N(\beta)$ the free-energy, internal energy and canonical entropy for a system with N 'particles'. The **free-energy density** is defined by

$$f(\beta) = \lim_{N\to\infty} F_N(\beta)/N\,, \tag{2.45}$$

if the limit exists, which is usually the case (at least if the forces between particles decrease fast enough at large distance). One defines analogously the **energy density** $u(\beta)$ and the **entropy density** $s(\beta)$.

The free-energy $F_N(\beta)$, is, quite generally, an analytic function of β in a neighborhood of the real β axis. This is a consequence of the fact that $Z(\beta)$ is analytic throughout the entire β plane, and strictly positive for real β's. A question of great interest is whether analyticity is preserved in the thermodynamic limit (2.45), under the assumption that the limit exists. Whenever the free-energy density $f(\beta)$ is non-analytic, one says that a **phase transition** occurs. Since the free-entropy density

$\phi(\beta) = -\beta f(\beta)$ is convex, the free-energy density is necessarily continuous whenever it exists.

In the simplest cases the non-analyticities occur at isolated points. Let β_c be such a point. Two particular type of singularities occur frequently:

- The free-energy density is continuous, but its derivative with respect to β is discontinuous at β_c. This singularity is named a **first order phase transition.**
- **The free-energy and its first derivative are continuous, but the second derivative is discontinuous at β_c. This is called a second order phase transition.**

Higher order phase transitions can be defined as well along the same lines.

Apart from being interesting mathematical phenomena, phase transitions correspond to *qualitative* changes in the underlying physical system. For instance the transition from water to vapor at 100°C at normal atmospheric pressure is modeled mathematically as a first order phase transition in the above sense. A great part of this book will be devoted to the study of phase transitions in many different systems, where the interacting 'particles' can be very diverse objects like information bits or occupation numbers on the vertices of a graph.

When N grows, the volume of the configuration space increases exponentially: $|\mathcal{X}_N| = |\mathcal{X}|^N$. Of course, not all the configurations are equally important under the Boltzmann distribution: lowest energy configurations have greater probability. What is important is therefore the number of configurations at a given energy. This information is encoded in the **energy spectrum** of the system:

$$\mathcal{N}_\Delta(E) = |\Omega_\Delta(E)|\,; \qquad \Omega_\Delta(E) \equiv \{x \in \mathcal{X}_N \,:\, E \le E(x) < E + \Delta\}\,. \tag{2.46}$$

In many systems of interest, the energy spectrum diverges exponentially as $N \to \infty$, if the energy is scaled linearly with N. More precisely, there exists a function $s(e)$ such that, given two numbers e and $\delta > 0$,

$$\lim_{N\to\infty} \frac{1}{N} \log \mathcal{N}_{N\delta}(Ne) = \sup_{e' \in [e, e+\delta]} s(e')\,. \tag{2.47}$$

The function $s(e)$ is called the **microcanonical entropy density**. The statement (2.47) is often rewritten in the more compact form:

$$\mathcal{N}_\Delta(E) \doteq_N \exp\left[Ns\left(\frac{E}{N}\right)\right]\,. \tag{2.48}$$

The notation $A_N \doteq_N B_N$ is used throughout the book to denote that two quantities A_N and B_N (which behave exponentially in N) are equal **to leading exponential order**, meaning: $\lim_{N\to\infty}(1/N)\log(A_N/B_N) = 0$. We often use $\doteq$ without index when there is no ambiguity on the large variable N.

The microcanonical entropy density $s(e)$ conveys a great amount of information about the system. Furthermore it is directly related to the intensive thermodynamic potentials through a fundamental relation:

Proposition 2.6 *If the microcanonical entropy density (2.47) exists for any e and if the limit in (2.47) is uniform in e, then the free-entropy density (2.45) exists and is given by:*

$$\phi(\beta) = \max_e [s(e) - \beta e] \,. \tag{2.49}$$

If the maximum of the $s(e) - \beta e$ is unique, then the internal energy density equals $\arg\max[s(e) - \beta e]$.

Proof: The basic idea is to write the partition function as follows

$$Z_N(\beta) \doteq \sum_{k=-\infty}^{\infty} \mathcal{N}_\Delta(k\Delta)\, e^{-\beta\Delta} \doteq \int \exp\{N s(e) - N\beta e\}\, \mathrm{d}e \,, \tag{2.50}$$

and to evaluate the last integral by saddle point. The reader will find references in the Notes section at the end of the chapter. □

Example 2.7 Let us consider N identical two-level systems: $\mathcal{X}_N = \mathcal{X} \times \cdots \times \mathcal{X}$, with $\mathcal{X} = \{1, 2\}$. We take the energy to be the sum of single-systems energies: $E(x) = E_{\text{single}}(x_1) + \cdots + E_{\text{single}}(x_N)$, with $x_i \in \mathcal{X}$. As in the previous Section we set $E_{\text{single}}(1) = \epsilon_1$, and $E_{\text{single}}(2) = \epsilon_2 > \epsilon_1$ and $\Delta = \epsilon_2 - \epsilon_1$.

The energy spectrum of this model is quite simple. For any energy $E = N\epsilon_1 + n\Delta$, there are $\binom{N}{n}$ configurations x with $E(x) = E$. Therefore, using the definition (2.47), we get

$$s(e) = \mathcal{H}\left(\frac{e - \epsilon_1}{\Delta}\right) . \tag{2.51}$$

Equation (2.49) can now be used to get

$$f(\beta) = \epsilon_1 - \frac{1}{\beta} \log(1 + e^{-\beta\Delta}) \,, \tag{2.52}$$

which agrees with the result obtained directly from the definition (2.18).

The great attention paid by physicists to the thermodynamic limit is extremely well justified by the huge number of degrees of freedom involved in a macroscopic piece of matter. Let us stress that the interest of the thermodynamic limit is more general than these huge numbers might suggest. First of all, it often happens that fairly small systems are well approximated by the thermodynamic limit. This is extremely important for numerical simulations of physical systems: one cannot of course simulate 10^{23} molecules on a computer! Even the cases in which the thermodynamic limit *is not* a good approximation are often fruitfully analyzed as *violations* of this limit. Finally, the insight gained in analyzing the $N \to \infty$ limit is always crucial in understanding moderate-size systems.

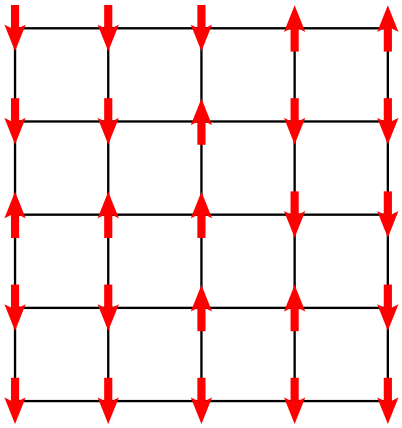

Fig. 2.2 A configuration of a two dimensional Ising model with $L = 5$. There is an Ising spin σ_i on each vertex i, shown by an arrow pointing up if $\sigma_i = +1$, pointing down if $\sigma_i = -1$. The energy (2.53) is given by the sum of two types of contributions: (i) A term $-\sigma_i\sigma_j$ for each edge (ij) of the graph, such that the energy is minimized when the two neighboring spins σ_i and σ_j point in the same direction; (ii) A term $-B\sigma_i$ for each site i, due to the coupling to an external magnetic field. The configuration depicted here has energy $-8 + 9B$

2.5 Ferromagnets and Ising models

Magnetic materials contain molecules with a magnetic moment, a three-dimensional vector which tends to align with the magnetic field felt by the molecule. Moreover, the magnetic moments of two distinct molecules interact with each other. Quantum mechanics plays an important role in magnetism. Because of quantum effects, the space of possible configurations of a magnetic moment becomes discrete. It is also at the origin of the so-called exchange interaction between magnetic moments. In many materials, the effect of the exchange interactions are such that the energy is lower when two moments align. While the behavior of a single magnetic moment in an external field is qualitatively simple, when we consider a bunch of interacting moments, the problem is much richer, and exhibits remarkable collective phenomena.

A simple mathematical model for such materials is the Ising model. It describes the magnetic moments by Ising spins localized at the vertices of a certain region of the d-dimensional cubic lattice. To keep things simple, let us consider a region $\mathbb{L}$ which is a cube of side L: $\mathbb{L} = \{1, \dots, L\}^d$. On each site $i \in \mathbb{L}$ there is an Ising spin $\sigma_i \in \{+1, -1\}$.

A configuration $\underline{\sigma} = (\sigma_1 \dots \sigma_N)$ of the system is given by assigning the values of all the spins in the system. Therefore the space of configurations $\mathcal{X}_N = \{+1, -1\}^{\mathbb{L}}$ has the form (2.1) with $\mathcal{X} = \{+1, -1\}$ and $N = L^d$.

The definition of ferromagnetic Ising models is completed by the definition of the energy function. A configuration $\underline{\sigma}$ has an energy:

$$E(\underline{\sigma}) = -\sum_{(ij)} \sigma_i\sigma_j - B\sum_{i\in\mathbb{L}} \sigma_i \,, \tag{2.53}$$

where the sum over (ij) runs over all the (unordered) couples of sites $i, j \in \mathbb{L}$ which are nearest neighbors. The real number B measures the applied external magnetic field.

Determining the free-energy density $f(\beta)$ in the thermodynamic limit for this model is a non-trivial task. The model was invented by Wilhem Lenz in the early twenties, who assigned the task of analyzing it to his student Ernst Ising. In his dissertation thesis (1924) Ising solved the $d = 1$ case and showed the absence of phase transitions. In 1948, Lars Onsager brilliantly solved the $d = 2$ case, exhibiting the first soluble "finite-dimensional" model with a second order phase transition. In higher dimensions the problem is unsolved although many important features of the solution are well understood.

Before embarking on any calculation, let us discuss some qualitative properties of this model. Two limiting cases are easily understood. At infinite temperature, $\beta = 0$, the energy (2.53) no longer matters and the Boltzmann distribution weights all the configurations with the same factor 2^{-N}. We have therefore an assembly of completely independent spins. At zero temperature, $\beta \to \infty$, the Boltzmann distribution concentrates onto the ground state(s). If there is no magnetic field, $B = 0$, there are two degenerate ground states: the configurations $\underline{\sigma}^{(+)}$ with all the spins pointing up, $\sigma_i = +1$, and the configuration $\underline{\sigma}^{(-)}$ with all the spins pointing down, $\sigma_i = -1$. If the magnetic field is set to some non-zero value, one of the two configuration dominates: $\underline{\sigma}^{(+)}$ if $B > 0$ and $\underline{\sigma}^{(-)}$ if $B < 0$.

Notice that the reaction of the system to the external magnetic field B is quite different in the two cases. To see this fact, define a "rescaled" magnetic field $x = \beta B$ and take the limits $\beta \to 0$ or $\beta \to \infty$ keeping x fixed. The expected value of any spin in $\mathbb{L}$, in the two limits, is:

$$\langle \sigma_i \rangle = \begin{cases} \tanh(x) & \text{for } \beta \to 0 \\ \tanh(Nx) & \text{for } \beta \to \infty \end{cases} . \tag{2.54}$$

Each spin reacts independently for $\beta \to 0$. On the contrary, they react as a whole as $\beta \to \infty$: one says that the response is cooperative.

A useful quantity for describing the response of the system to the external field is the **average magnetization**:

$$M_N(\beta, B) = \frac{1}{N} \sum_{i \in \mathbb{L}} \langle \sigma_i \rangle . \tag{2.55}$$

Because of the symmetry between the up and down directions, $M_N(\beta, B)$ is an odd function of B. In particular $M_N(\beta, 0) = 0$. A cooperative response can be emphasized by considering the **spontaneous magnetization**

$$M_+(\beta) = \lim_{B \to 0+} \lim_{N \to \infty} M_N(\beta, B) . \tag{2.56}$$

It is important to understand that a non-zero spontaneous magnetization can appear only in an infinite system: the order of the limits in Eq. (2.56) is crucial. Our analysis so far has shown that the spontaneous magnetization exists at $\beta = \infty$: $M_+(\infty) = 1$. On the other hand $M_+(0) = 0$. It can be shown that actually the spontaneous

magnetization $M(\beta)$ is always zero in a high temperature phase $\beta < \beta_c(d)$ (such a phase is called **paramagnetic**). In one dimension ($d = 1$), we will show below that $\beta_c(1) = \infty$. The spontaneous magnetization is always zero, except at zero temperature ($\beta = \infty$): one speaks of a zero temperature phase transition. In dimensions $d \geq 2$, $\beta_c(d)$ is finite, and $M(\beta)$ becomes non zero in the so called **ferromagnetic phase** $\beta > \beta_c$: a phase transition takes place at $\beta = \beta_c$. The temperature $T_c = 1/\beta_c$ is called the **critical temperature**. In the following we shall discuss the $d = 1$ case, and a variant of the model, called the Curie Weiss model, where each spin interacts with all the other ones: this is the simplest model which exhibits a finite temperature phase transition.

2.5.1 The one-dimensional case

The $d = 1$ case has the advantage of being simple to solve. We want to compute the partition function (2.4) for a system of N spins with energy $E(\underline{\sigma}) = -\sum_{i=1}^{N-1} \sigma_i \sigma_{i+1} - B \sum_{i=1}^{N} \sigma_i$. We will use the so-called **transfer matrix method**, which belongs to the general dynamic programming strategy familiar to computer scientists.

We introduce the partial partition function where the configurations of all spins $\sigma_1, \ldots, \sigma_p$ have been summed over, at fixed σ_{p+1}:

$$z_p(\beta, B, \sigma_{p+1}) \equiv \sum_{\sigma_1, \ldots, \sigma_p} \exp\left[\beta \sum_{i=1}^{p} \sigma_i \sigma_{i+1} + \beta B \sum_{i=1}^{p} \sigma_i\right] . \tag{2.57}$$

The partition function (2.4) is given by $Z_N(\beta, B) = \sum_{\sigma_N} z_{N-1}(\beta, B, \sigma_N) \exp(\beta B \sigma_N)$. Obviously z_p satisfies the recursion relation

$$z_p(\beta, B, \sigma_{p+1}) = \sum_{\sigma_p = \pm 1} T(\sigma_{p+1}, \sigma_p) z_{p-1}(\beta, B, \sigma_p) \tag{2.58}$$

where we define the **transfer matrix** $T(\sigma, \sigma') = \exp[\beta\sigma\sigma' + \beta B\sigma']$. This is the 2×2 matrix:

$$T = \begin{pmatrix} e^{\beta+\beta B} & e^{-\beta-\beta B} \\ e^{-\beta+\beta B} & e^{\beta-\beta B} \end{pmatrix} \tag{2.59}$$

Introducing the two component vectors $\psi_L = \begin{pmatrix} \exp(\beta B) \\ \exp(-\beta B) \end{pmatrix}$ and $\psi_R = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, and the standard scalar product between vectors $(a, b) = a_1 b_1 + a_2 b_2$, the partition function can be written in matrix form:

$$Z_N(\beta, B) = (\psi_L, T^{N-1} \psi_R) . \tag{2.60}$$

Let us call λ_1, λ_2 the eigenvalues of T, and ψ_1, ψ_2 the corresponding eigenvectors. It is easy to realize that ψ_1, ψ_2 can be chosen to be linearly independent, hence ψ_R can be decomposed as $\psi_R = u_1 \psi_1 + u_2 \psi_2$. The partition function is then expressed as:

$$Z_N(\beta, B) = u_1 \, (\psi_L, \psi_1) \, \lambda_1^{N-1} + u_2 \, (\psi_L, \psi_2) \, \lambda_2^{N-1} . \tag{2.61}$$

The diagonalization of the matrix T gives:

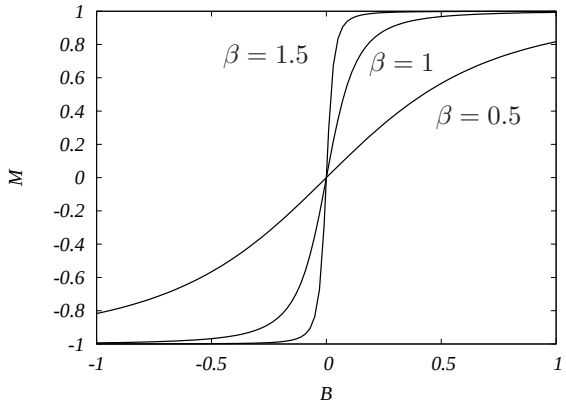

Fig. 2.3 The average magnetization of the one dimensional Ising model, as a function of the magnetic field B, at inverse temperatures $\beta = 0.5, 1, 1.5$

$$\lambda_{1,2} = e^{\beta}\cosh(\beta B) \pm \sqrt{e^{2\beta}\sinh^2\beta B + e^{-2\beta}}\,. \tag{2.62}$$

For β finite, in the large N limit, the partition function is dominated by the largest eigenvalue λ_1, and the free entropy density is given by $\phi = \log\lambda_1$:

$$\phi(\beta, B) = \log\left[e^{\beta}\cosh(\beta B) + \sqrt{e^{2\beta}\sinh^2\beta B + e^{-2\beta}}\right]. \tag{2.63}$$

Using the same transfer matrix technique we can compute expectation values of observables. For instance the expected value of a given spin is

$$\langle\sigma_i\rangle = \frac{1}{Z_N(\beta, B)}\,(\psi_{\rm L}, T^{i-1}\hat{\sigma}T^{N-i}\psi_{\rm R})\,, \tag{2.64}$$

where $\hat{\sigma}$ is the following matrix:

$$\hat{\sigma} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2.65}$$

Averaging over the position i, one can compute the average magnetization $M_N(\beta, B)$. In the thermodynamic limit we get

$$\lim_{N\to\infty} M_N(\beta, B) = \frac{\sinh\beta B}{\sqrt{\sinh^2\beta h + e^{-4\beta}}} = \frac{1}{\beta}\frac{\partial\phi}{\partial B}(\beta, B)\,. \tag{2.66}$$

Both the free-energy and the average magnetization turn out to be analytic functions of β and B for $\beta < \infty$. In particular the spontaneous magnetization vanishes at any non-zero temperature:

$$M_+(\beta) = 0\,, \qquad \forall\beta < \infty\,. \tag{2.67}$$

In Fig. 2.3 we plot the average magnetization $M(\beta, B) \equiv \lim_{N\to\infty} M_N(\beta, B)$ as a function of the applied magnetic field B for various values of the temperature β. The

curves become steeper and steeper as β increases. This statement can be made more quantitative by computing the **susceptibility** associated to the average magnetization:

$$\chi_M(\beta) = \frac{\partial M}{\partial h}(\beta, 0) = \beta\, e^{2\beta}\,. \tag{2.68}$$

This result can be interpreted as follows. A single spin in a field has susceptibility $\chi(\beta) = \beta$. If we consider N spins constrained to take the same value, the corresponding susceptibility will be $N\beta$, as in Eq. (2.54). In the present case the system behaves as if the spins were blocked into groups of $\chi_M(\beta)/\beta$ spins each. The spins in each group are constrained to take the same value, while spins belonging to different blocks are independent.

This qualitative interpretation receives further support by computing a **correlation function**.

Exercise 2.6 Consider the one dimensional Ising model in zero field, $B = 0$. Show that, when $\delta N < i < j < (1-\delta)N$, the correlations function $\langle \sigma_i \sigma_j \rangle$ is, in the large N limit:

$$\langle \sigma_i \sigma_j \rangle = e^{-|i-j|/\xi(\beta)} + \Theta(e^{-\alpha N})\,, \tag{2.69}$$

with $\xi(\beta) = -1/\log\tanh\beta$.
[Hint: You can either use the general transfer matrix formalism, or more simply use the identity $e^{\beta\sigma_i\sigma_{i+1}} = \cosh\beta(1 + \sigma_i\sigma_{i+1}\tanh\beta)$]

Notice that, in Eq.(2.69), $\xi(\beta)$ gives the typical distance below which two spins in the system are well correlated. For this reason it is usually called the **correlation length** of the model. This correlation length increases when the temperature decreases: spins become correlated at larger and larger distances. The result (2.69) is clearly consistent with our interpretation of the susceptibility. In particular, as $\beta \to \infty$, $\xi(\beta) \approx e^{2\beta}/2$ and $\chi_M(\beta) \approx 2\beta\xi(\beta)$.

The connection between correlation length and susceptibility is very general and can be understood as a consequence of the fluctuation-dissipation theorem (2.44):

$$\begin{aligned}\chi_M(\beta) &= \beta N \left\langle \left(\frac{1}{N}\sum_{i=1}^N \sigma_i\right) ; \left(\frac{1}{N}\sum_{i=1}^N \sigma_i\right) \right\rangle \\ &= \frac{\beta}{N}\sum_{i,j=1}^N \langle \sigma_i\,;\,\sigma_j\rangle = \frac{\beta}{N}\sum_{i,j=1}^N \langle \sigma_i\sigma_j\rangle\,,\end{aligned} \tag{2.70}$$

where the last equality comes from the fact that $\langle \sigma_i \rangle = 0$ when $B = 0$. Using (2.69), we get

$$\chi_M(\beta) = \beta \sum_{i=-\infty}^{+\infty} e^{-|i|/\xi(\beta)} + \Theta(e^{-\alpha N})\,. \tag{2.71}$$

It is therefore evident that a large susceptibility must correspond to a large correlation length.

2.5.2 The Curie-Weiss model

The exact solution of the one-dimensional model lead Ising to think that there couldn't be a phase transition in any dimension. Some thirty years earlier a qualitative theory of ferromagnetism had been put forward by Pierre Curie. Such a theory assumed the existence of a phase transition at non-zero temperature T_c (the so-called the "Curie point") and a non-vanishing spontaneous magnetization for $T < T_c$. The dilemma was eventually solved by Onsager solution of the two-dimensional model.

Curie theory is realized exactly within a rather abstract model: the so-called **Curie-Weiss model**. We shall present it here as one of the simplest solvable models with a finite-temperature phase transition. Once again we have N Ising spins $\sigma_i \in \{\pm 1\}$ and a configuration is given by $\underline{\sigma} = (\sigma_1, \dots, \sigma_N)$. However the spins no longer sits on a d-dimensional lattice: they all interact in pairs. The energy function, in presence of a magnetic field B, is given by:

$$E(\underline{\sigma}) = -\frac{1}{N} \sum_{(ij)} \sigma_i \sigma_j - B \sum_{i=1}^{N} \sigma_i \,, \tag{2.72}$$

where the sum on (ij) runs over all the $N(N-1)/2$ couples of spins. Notice the peculiar $1/N$ scaling in front of the exchange term. The exact solution presented below shows that this is the only choice which yields a non-trivial free-energy density in the thermodynamic limit. This can be easily understood intuitively as follows. The sum over (ij) involves $\Theta(N^2)$ terms of order $\Theta(1)$. In order to get an energy function scaling as N, we need to put a $1/N$ coefficient in front.

In adopting the energy function (2.72), we gave up the description of any finite-dimensional geometrical structure. This is a severe simplification, but has the advantage of making the model exactly soluble. The Curie-Weiss model is the first example of a large family: the so-called **mean-field models**. We will explore many instances of this family throughout the book.

A possible approach to the computation of the partition function consists in observing that the energy function can be written in terms of a simple observable, the **instantaneous** (or, **empirical**) **magnetization**:

$$m(\underline{\sigma}) = \frac{1}{N} \sum_{i=1}^{N} \sigma_i \,. \tag{2.73}$$

Notice that this is a function of the configuration $\underline{\sigma}$, and shouldn't be confused with its expected value, the average magnetization, cf. Eq. (2.55). It is a "simple" observable because it is equal to the sum of observables depending upon a single spin.

We can write the energy of a configuration in terms of its instantaneous magnetization:

$$E(\underline{\sigma}) = \frac{1}{2} N - \frac{1}{2} N\, m(\underline{\sigma})^2 - NB\, m(\underline{\sigma}) \,. \tag{2.74}$$

This implies the following formula for the partition function

$$Z_N(\beta, B) = e^{-N\beta/2} \sum_m \mathcal{N}_N(m) \, \exp\left\{ \frac{N\beta}{2} m^2 + N\beta B m \right\} , \tag{2.75}$$

where the sum over m runs over all the possible instantaneous magnetizations of N Ising spins: $m = -1 + 2k/N$ with $0 \le k \le N$ an integer number, and $\mathcal{N}_N(m)$ is the number of configurations having a given instantaneous magnetization m. This is a binomial coefficient whose large-N behavior is expressed in terms of the entropy function of Bernoulli variables:

$$\mathcal{N}_N(m) = \binom{N}{N\frac{1+m}{2}} \doteq \exp\left[N \, \mathcal{H}\left(\frac{1+m}{2} \right) \right] . \tag{2.76}$$

To leading exponential order in N, the partition function can thus be written as:

$$Z_N(\beta, B) \doteq \int_{-1}^{+1} e^{N \phi_{\rm mf}(m;\beta,B)} \, \mathrm{d}m \, , \tag{2.77}$$

where we have defined

$$\phi_{\rm mf}(m; \beta, B) = -\frac{\beta}{2}(1 - m^2) + \beta B m + \mathcal{H}\left(\frac{1+m}{2} \right) . \tag{2.78}$$

The integral in (2.77) is easily evaluated by Laplace method, to get the final result for the free-energy density

$$\phi(\beta, B) = \max_{m \in [-1,+1]} \phi_{\rm mf}(m; \beta, B) \, . \tag{2.79}$$

One can see that the maximum is obtained away from the boundary points, so that the corresponding m must be a stationary point of $\phi_{\rm mf}(m; \beta, B)$, which satisfies the **saddle-point equation** $\partial \phi_{\rm mf}(m; \beta, B)/\partial m = 0$:

$$m_* = \tanh(\beta m_* + \beta B) \, . \tag{2.80}$$

In the above derivation we were slightly sloppy at two steps: substituting the binomial coefficient with its asymptotic form and changing the sum over m into an integral. The mathematically minded reader is invited to show that these passages are indeed correct. ⋆

With a bit more work, the above method can be extended to expectation values of observables. Let us consider for instance the average magnetization $M(\beta, B)$. It can be ⋆ easily shown that, whenever the maximum of $\phi_{\rm mf}(m; \beta, B)$ over m is non-degenerate,

$$M(\beta, B) \equiv \lim_{N\to\infty} \langle m(\underline{\sigma}) \rangle = m_*(\beta, B) \equiv \arg\max_m \phi_{\rm mf}(m; \beta, B) \, , \tag{2.81}$$

We can now examine the implications that can be drawn from Eqs. (2.79) and (2.80). Let us first consider the $B = 0$ case (see Fig.2.4). The function $\phi_{\rm mf}(m; \beta, 0)$ is symmetric in m. For $0 \le \beta \le 1 \equiv \beta_{\rm c}$, it is also concave and achieves its unique maximum in $m_*(\beta) = 0$. For $\beta > 1$, $m = 0$ remains a stationary point but becomes a local minimum, and the function develops two degenerate global maxima at $m_\pm(\beta)$

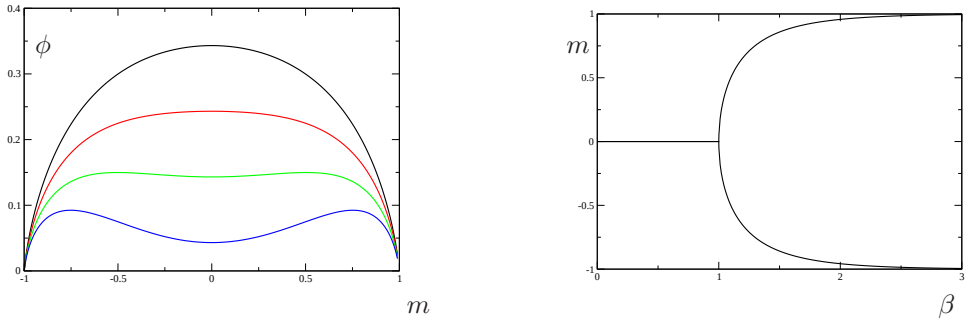

Fig. 2.4 Left: the function $\phi_{\rm mf}(m;\beta,B=0)$ is plotted versus m, for $\beta = 0.7, 0.9, 1.1, 1.3$ (from top to bottom). For $\beta < \beta_{\rm c} = 1$ there is a unique maximum at $m = 0$, for $\beta < \beta_{\rm c} = 1$ there are two degenerate maxima at two symmetric values $\pm m_+(\beta)$. Right: values of m which maximize $\phi_{\rm mf}(m;\beta,B=0)$ are plotted versus β. The phase transition at $\beta_{\rm c} = 1$ is signaled by the bifurcation.

with $m_+(\beta) = -m_-(\beta) > 0$. These two maxima bifurcate continuously from $m = 0$ at $\beta = \beta_{\rm c}$.

A phase transition takes place at $\beta_{\rm c}$. Its meaning can be understood by computing the expectation value of the spins. Notice that the energy function (2.72) is symmetric under a spin-flip transformation which maps $\sigma_i \to -\sigma_i$ for all i's. Therefore $\langle \sigma_i \rangle = \langle (-\sigma_i) \rangle = 0$ and the average magnetization vanishes $M(\beta, 0) = 0$. On the other hand, the spontaneous magnetization, defined in (2.56), is zero in the paramagnetic phase $\beta < \beta_{\rm c}$, and equal to $m_+(\beta)$ in the ferromagnetic phase $\beta > \beta_{\rm c}$. The physical interpretation of this phase is the following: for any finite N the pdf of the instantaneous magnetization $m(\underline{\sigma})$ has two symmetric peaks, at $m_\pm(\beta)$, which become sharper and sharper as N increases. Any external perturbation which breaks the symmetry between the peaks, for instance a small positive magnetic field B, favors one peak with respect to the other one, and therefore the system develops a spontaneous magnetization. Let us stress that the occurrence of a phase transition is a property of systems in the thermodynamic limit $N \to \infty$.

In physical magnets, symmetry breaking can come for instance from impurities, subtle effects of dipolar interactions together with the shape of the magnet, or an external magnetic field. The result is that at low enough temperatures some systems, the ferromagnets, develop a spontaneous magnetization. If you heat a magnet made of iron, its magnetization disappears at a critical temperature $T_{\rm c} = 1/\beta_{\rm c} \approx 770$ degrees Celsius. The Curie Weiss model is a simple solvable case exhibiting this phase transition.

Exercise 2.7 Compute the expansion of $m_+(\beta)$ and of $\phi(\beta, B = 0)$ near $\beta = \beta_{\rm c}$, and show that the transition is of second order. Compute the low temperature behavior of the spontaneous magnetization.

Exercise 2.8 Inhomogeneous Ising chain. The one-dimensional Ising problem does not have a finite temperature phase transition, as long as the interactions are short range and translational invariant. On the other hand, if the couplings in the Ising chain grow fast enough at large distance, one can have a phase transition. This is not a very realistic model from the point of view of physics, but it is useful as a solvable example of phase transition.

Consider a chain of Ising spins $\sigma_0, \sigma_1, \ldots, \sigma_N$ with energy $E(\underline{\sigma}) = -\sum_{n=0}^{N-1} J_n \sigma_n \sigma_{n+1}$. Suppose that the coupling constants J_n form a positive, monotonously increasing sequence, growing logarithmically. More precisely, we assume that $\lim_{n\to\infty} J_n/\log n = 1$. Denote by $\langle\cdot\rangle_+$ (resp. $\langle\cdot\rangle_-$) the expectation value with respect to Boltzmann's probability distribution when the spin σ_N is fixed to $\sigma_N = +1$ (resp. fixed to $\sigma_N = -1$).

(*i*) Show that , for any $n \in \{0\ldots, N-1\}$, the magnetization is $\langle\sigma_n\rangle_\pm = \prod_{p=n}^{N-1} \tanh(\beta J_p)$

(*ii*) Show that the critical inverse temperature $\beta_c = 1/2$ separates two regimes, such that: for $\beta < \beta_c$, one has $\lim_{N\to\infty}\langle\sigma_n\rangle_+ = \lim_{N\to\infty}\langle\sigma_n\rangle_- = 0$; for $\beta > \beta_c$, one has $\lim_{N\to\infty}\langle\sigma_n\rangle_\pm = \pm M(\beta)$, with $M(\beta) > 0$.

Notice that in this case, the role of the symmetry breaking field is played by the choice of boundary condition.

2.6 The Ising spin glass

In real magnetic materials, localized magnetic moments are subject to several sources of interactions. Apart from the exchange interaction mentioned in the previous Section, they may interact through intermediate conduction electrons, etc... As a result, depending on the material which one considers, their interaction can be either ferromagnetic (their energy is minimized when they are parallel) or **antiferromagnetic** (their energy is minimized when they point *opposite* to each other). **Spin glasses** are a family of materials whose magnetic properties are particularly complex. They can be produced by diluting a small fraction of a 'transition magnetic metal' like manganese into a 'noble metal' like copper in a ratio, say, of 1 : 100. In such an alloy, magnetic moments are localized at manganese atoms, which are placed at random positions in a copper background. Depending on the distance of two manganese atoms, the net interaction between their magnetic moments can be either ferromagnetic or antiferromagnetic.

The **Edwards-Anderson model** is a widely accepted mathematical abstraction of these physical systems. Once again, the basic degrees of freedom are Ising spins $\sigma_i \in \{-1, +1\}$ sitting on the vertices of a d-dimensional cubic lattice $\mathbb{L} = \{1, \ldots, L\}^d$, $i \in \mathbb{L}$. The configuration space is therefore $\{-1, +1\}^{\mathbb{L}}$. As in the ferromagnetic Ising model, the energy function reads

$$E(\underline{\sigma}) = -\sum_{(ij)} J_{ij}\sigma_i\sigma_j - B\sum_{i\in\mathbb{L}} \sigma_i\,, \tag{2.82}$$

where $\sum_{(ij)}$ runs over each edge of the lattice. Unlike in the Ising ferromagnet, a different coupling constant J_{ij} is now associated to each edge (ij), and its sign can be positive or negative. The interaction between spins σ_i and σ_j is ferromagnetic if $J_{ij} > 0$ and antiferromagnetic if $J_{ij} < 0$.

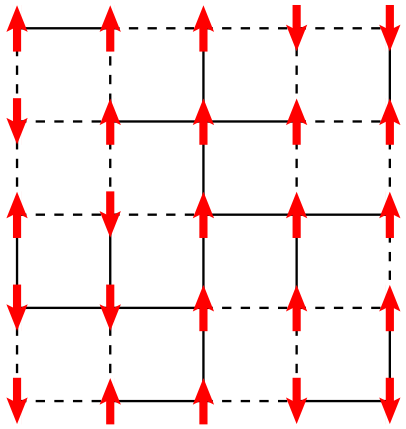

Fig. 2.5 A configuration of a two dimensional Edwards-Anderson model with $L = 5$. Spins are coupled by two types of interactions: ferromagnetic ($J_{ij} = +1$), indicated by a continuous line, and antiferromagnetic ($J_{ij} = -1$), indicated by a dashed line. The energy of the configuration shown here is $-14 - 7h$.

A pictorial representation of this energy function is given in Fig. 2.5. The Boltzmann distribution is given by

$$\mu_\beta(\underline{\sigma}) = \frac{1}{Z(\beta)} \exp\left\{\beta \sum_{(ij)} J_{ij}\sigma_i\sigma_j + \beta B \sum_{i\in\mathbb{L}} \sigma_i\right\}, \tag{2.83}$$

$$Z(\beta) = \sum_{\underline{\sigma}} \exp\left\{\beta \sum_{(ij)} J_{ij}\sigma_i\sigma_j + \beta B \sum_{i\in\mathbb{L}} \sigma_i\right\}. \tag{2.84}$$

It is important to notice that the couplings $\{J_{ij}\}$ play a completely different role from the spins $\{\sigma_i\}$. The couplings are just parameters involved in the definition of the energy function, as the magnetic field B, and they are not summed over when computing the partition function. In principle, for any particular sample of a magnetic material, one should estimate experimentally the values of the J_{ij}'s, and then compute the partition function. We could have made explicit the dependence of the partition function and of the Boltzmann distribution on the couplings by using notations such as $Z(\beta, B; \{J_{ij}\})$, $\mu_{\beta,B;\{J_{ij}\}}(\underline{\sigma})$. However, when these explicit mentions are not necessary, we prefer to keep to lighter notations.

The present understanding of the Edwards-Anderson model is much poorer than for the ferromagnetic models introduced in the previous Section. The basic reason of this difference is **frustration** and is illustrated in Fig. 2.6 on an $L = 2$, $d = 2$ model (a model consisting of just 4 spins).

A spin glass is frustrated whenever there exist local constraints that are in conflict, meaning that it is not possible to satisfy all of them simultaneously. In the Edwards-Anderson model, a plaquette is a group of four neighboring spins forming a square (i.e.

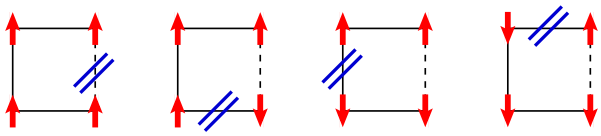

Fig. 2.6 Four configurations of a small Edwards-Anderson model: continuous lines indicate ferromagnetic interactions ($J_{ij} = +1$), while dashed lines are for antiferromagnetic interactions ($J_{ij} = -1$). In zero magnetic field ($B = 0$), the four configurations are degenerate and have energy $E = -2$. The double bar indicates an unsatisfied interaction. Notice that there is no configuration with lower energy. This system is frustrated since it is impossible to satisfy simultaneously all constraints.

a cycle of length four). A plaquette is frustrated if and only if the product of the J_{ij} along all four edges of the plaquette is negative. As shown in Fig. 2.6, it is then impossible to minimize simultaneously all the four local energy terms associated with each edge. In a spin glass, the presence of a finite density of frustrated plaquettes generates a very complicated energy landscape. The resulting effect of all the interactions is not obtained by 'summing' the effects of each of them separately, but is the outcome of a complex interplay. The ground state spin configuration (the one satisfying the largest possible number of interactions) is difficult to find: it cannot be guessed on symmetry grounds. It is also frequent to find in a spin glass a configuration which is very different from the ground state but has an energy very close to the ground state energy. We shall explore these and related issues throughout the book.

Notes

There are many good introductory textbooks on statistical physics and thermodynamics, for instance the books (Reif, 1965) or (Huang, 1987). Going towards more advanced texts, one can suggest the books (Ma, 1985) and (Parisi, 1988). A more mathematically minded presentation can be found in the books (Galavotti, 1999) and (Ruelle, 1999). The reader will find there the proof of Proposition 2.6.

The two-dimensional Ising model at vanishing external field can also be solved by a transfer matrix technique, see for instance (Baxter, 1982). The transfer matrix, which passes from a column of the lattice to the next, is a $2^L \times 2^L$ matrix, and its dimension diverges exponentially with the lattice size L. Finding its largest eigenvalue is therefore a complicated task. Nobody has found the solution so far for $B \neq 0$.

Spin glasses will be a recurring theme in this book, and more will be said about them in the next Chapters. An introduction to this subject from a physicist point of view is provided by the book (Fischer and Hetz, 1993) or the review (Binder and Young, 1986). The concept of frustration was introduced by (Toulouse, 1977).

3

Introduction to combinatorial optimization

This Chapter provides an elementary introduction to some basic concepts in theoretical computer science. Which computational tasks can/cannot be accomplished efficiently by a computer? How much resources (time, memory, etc.) are needed for solving a specific problem? What are the performances of a specific solution method (an algorithm), and, whenever more than one method is available, which one is preferable? Are some problems intrinsically harder than others? This are some of the questions one would like to answer.

One large family of computational problems is formed by combinatorial optimization problems. These consist in finding an element of a finite set which maximizes (or minimizes) an easy-to-evaluate objective function. Several features make such problems particularly interesting. First of all, most of the times they are equivalent to decision problems (questions which require a YES/NO answer), which is the most fundamental class of problems within computational complexity theory. Second, optimization problems are ubiquitous both in applications and in pure sciences. In particular, there exist some evident connections both with statistical mechanics and with coding theory. Finally, they form a very large and well studied family, and therefore an ideal context for understanding some advanced issues. One should however keep in mind that computation is more than just combinatorial optimization. A larger family, that we will also discuss later on, contains the counting problems: one wants to count how many elements of a finite set have some easy-to-check property. Finally, there are other important families of computational problem that we shall not address at all, like continuous optimization problems.

The study of combinatorial optimization is introduced in Sec. 3.1 through the simple example of the minimum spanning tree. This section also contains the basic definitions of graph theory that we use throughout the book. General definitions and terminology are given in Sec. 3.2. These definitions are further illustrated in Sec. 3.3 through several additional examples. Section 3.4 provides an informal introduction to some basic concepts in computational complexity: we define the classes P and NP, and the notion of NP-completeness. As mentioned above, combinatorial optimization problems often appear in pure sciences and applications. The examples of statistical physics and coding are briefly discussed in Secs. 3.5 and 3.6.

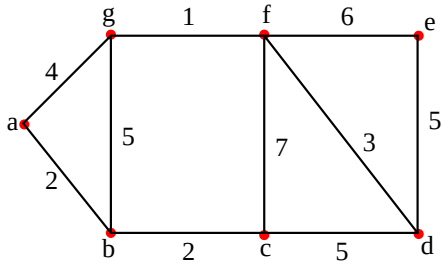

Fig. 3.1 This graph has 7 vertices (labeled a to g) and 10 edges. The 'cost' of each edge is indicated next to it. In the Minimum Spanning Tree problem, one seeks a loop-free subgraph of minimum cost connecting all vertices.

3.1 A first example: minimum spanning tree

The minimum spanning tree problem is easily stated and may appear in many practical applications. Suppose for instance you have a bunch of computers in a building. You may want to connect them pairwise in such a way that the resulting network is connected and the amount of cable used is minimum.

3.1.1 Definition and basics of graph theory

A mathematical abstraction of the above practical problem requires a few basic definitions from graph theory. A **graph** is a set $\mathcal{V}$ of vertices, labeled by $\{1, 2, \ldots, |\mathcal{V}|\}$, together with a set $\mathcal{E}$ of edges connecting them: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The vertex set can be any finite set but one often takes the set of the first $|\mathcal{V}|$ integers: $\mathcal{V} = \{1, 2, \ldots, |\mathcal{V}|\}$. The edges are simply unordered couples of distinct vertices $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. For instance an edge joining vertices i and j is identified as $e = (i, j)$. A **weighted graph** is a graph where a cost (a real number) is associated with every edge. The **degree** of a vertex is the number of edges connected to it. A **path** between two vertices i and j is a set of edges $\{(j, i_2); (i_2, i_3); (i_3, i_4); \ldots; (i_{r-1}, i_r); (i_r, j)\} \subseteq \mathcal{E}$. A graph is **connected** if, for every pair of vertices, there is a path which connects them. A **completely connected** graph, or **complete** graph, also called a **clique**, is a graph where all the $|\mathcal{V}|(|\mathcal{V}|-1)/2$ edges are present. A **cycle** is a path starting and ending on the same vertex. A **tree** is a connected graph without cycles.

Consider the graph in Fig. 3.1. You are asked to find a tree (a subset of the edges forming a cycle-free subgraph) such that any two vertices are connected by exactly one path (in this case the tree is said to be spanning). To find such a subgraph is an easy task. The edges $\{(a, b); (b, c); (c, d); (b, g); (d, e)\}$, for instance, do the job. However in our problem a cost is associated with each edge. The cost of a subgraph is assumed to be equal to the sum of the costs of its edges, and you want to minimize it. This is a non-trivial problem.

In general, an instance of the **minimum spanning tree** (MST) problem is given by a connected weighted graph (each edge e has a cost $w(e) \in \mathbb{R}$). The optimization problem consists in finding a spanning tree with minimum cost. What one seeks is an

algorithm which, given an instance of the MST problem, outputs the spanning tree with lowest cost.

3.1.2 An efficient algorithm

The simple-minded approach would consist in enumerating all the spanning trees for the given graph, and comparing their weights. However the number of spanning trees grows very rapidly with the size of the graph. Consider, as an example, the complete graph on N vertices. The number of spanning trees of such a graph is, according to the Cayley formula, N^{N-2}. Even if the cost of any such tree were evaluated in 10^{-3} sec, it would take 2 years to find the MST of a $N = 12$ graph, and half a century for $N = 13$. At the other extreme, if the graph is very simple, it may contain a small number of spanning trees, a single one in the extreme case where the graph is itself a tree. Nevertheless, in most interesting examples the situation is nearly as dramatic as in the complete graph case.

A much better algorithm can be obtained from the following theorem:

Theorem 3.1 *Let $\mathcal{U} \subset \mathcal{V}$ be a proper subset of the vertex set $\mathcal{V}$ (such that neither $\mathcal{U}$ nor $\mathcal{V}\backslash\mathcal{U}$ are empty). Let us consider the subset $\mathcal{F}$ of edges which connect a vertex in $\mathcal{U}$ to a vertex in $\mathcal{V}\backslash\mathcal{U}$, and let $e \in \mathcal{F}$ be an edge of lowest cost in this subset: $w(e) \leq w(e')$ for any $e' \in \mathcal{F}$. If there are several such edges, e can be any one of them. Then there exists a minimum spanning tree which contains e.*

Proof: Consider a MST $\mathcal{T}$, and suppose that it does not contain the edge e mentioned in the statement. This edge is such that $e = (i, j)$ with $i \in \mathcal{U}$ and $j \in \mathcal{V}\backslash\mathcal{U}$. The spanning tree $\mathcal{T}$ must contain a path between i and j. This path contains at least one edge f connecting a vertex in $\mathcal{U}$ to a vertex in $\mathcal{V}\backslash\mathcal{U}$, and f is distinct from e. Now consider the subgraph $\mathcal{T}'$ built from $\mathcal{T}$ by removing the edge f and adding the edge e. We leave to the reader the exercise of showing that $\mathcal{T}'$ is a spanning tree. If we denote by $E(\mathcal{T})$ the cost of tree $\mathcal{T}$, $E(\mathcal{T}') = E(\mathcal{T}) + w(e) - w(f)$. Since $\mathcal{T}$ is a MST, $E(\mathcal{T}') \geq E(\mathcal{T})$. On the other hand e has minimum cost within $\mathcal{F}$, hence $w(e) \leq w(f)$. Therefore $w(e) = w(f)$ and $\mathcal{T}'$ is a MST containing e. □

This result allows to construct a minimum spanning tree of $\mathcal{G}$ incrementally. One starts from a single vertex. At each step a new edge is added to the tree, whose cost is minimum among all the ones connecting the already existing tree with the remaining vertices. After $N - 1$ iterations, the tree will be spanning.

MST ALGORITHM (Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, weight function $w : \mathcal{E} \to \mathbb{R}_+$)

Set $\mathcal{U} := \{1\}$, $\mathcal{T} := \emptyset$ and $E = 0$;
while $\mathcal{V}\backslash\mathcal{U}$ is not empty:
    Let $\mathcal{F} := \{e = (i, j) \in \mathcal{E}$ such that $i \in \mathcal{U},\ j \in \mathcal{V}\backslash\mathcal{U}\}$;
    Find $e_* = (i_*, j_*) := \arg\min_{e\in\mathcal{F}}\{w(e)\}$;
    Set $\mathcal{U} := \mathcal{U} \cup j_*$, $\mathcal{T} := \mathcal{T} \cup e_*$, and $E := E + w(e_*)$;
end
return the spanning tree $\mathcal{T}$ and its cost E.

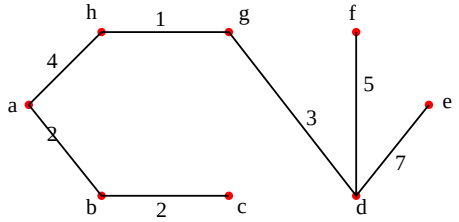

Fig. 3.2 A minimum spanning tree for the graph defined in Fig. 3.1. The cost of this tree is $E = 17$.

Exercise 3.1 Write a code for this algorithm, and find a MST for the problem described in Fig. 3.1. A solution is given in Fig. 3.2

Exercise 3.2 Show explicitly that the algorithm MST always outputs a minimum spanning tree.

Theorem 3.1 establishes that, for any $\mathcal{U} \subset \mathcal{V}$, and any lowest cost edge e among the ones connecting $\mathcal{U}$ to $\mathcal{V}\backslash\mathcal{U}$, there exists a MST containing e. This does not guarantee that, when two different sets $\mathcal{U}_1$ and $\mathcal{U}_2$, and the corresponding lowest cost edges e_1 and e_2 are considered, there exists a MST containing *both* e_1 and e_2. The above algorithm works by constructing a sequence of such $\mathcal{U}$'s and adding to the tree the corresponding lowest weight edges. It is therefore not obvious a priori that it will output a MST (unless this is unique).

Let us analyze the number of elementary operations required by the algorithm to construct a spanning tree on an N nodes graph. By 'elementary operation' we mean comparisons, sums, multiplications, etc, all of them counting as one. Of course, the number of such operations depends on the graph, but we can find a simple upper bound by considering the completely connected graph. Most of the operations in the above algorithm are comparisons among edge weights for finding e_* in step 4. In order to identify e_*, one has to scan at most $|\mathcal{U}| \times |\mathcal{V}\backslash\mathcal{U}| = |\mathcal{U}| \times (N - |\mathcal{U}|)$ edges connecting $\mathcal{U}$ to $\mathcal{V}\backslash\mathcal{U}$. Since $|\mathcal{U}| = 1$ at the beginning and is augmented of one element at each iteration of the cycle 2-6, the number of comparisons is upper bounded by $\sum_{U=0}^{N} U(N-U) \leq N^3/6$[1]. This is an example of a polynomial algorithm, whose computing time grows like a power of the number of vertices. The insight gained from the theorem provides an algorithm which is much better than the naive one, at least when N gets large.

[1]The algorithm can be easily improved by keeping an ordered list of the edges already encountered

Exercise 3.3 Suppose you are given a weighted graph $(\mathcal{V}, \mathcal{E})$ in which the weights are all different, and the edges are ordered in such a way that their weights form an increasing sequence $w(e_1) < w(e_2) < w(e_3) < \cdots$. Another graph with the same $(\mathcal{V}, \mathcal{E})$ has different weights $w'(e)$, but they are also increasing along the same sequence $w'(e_1) < w'(e_2) < w'(e_3) < \cdots$. Show that the MST is the same in these two graphs.

3.2 General definitions

MST is an example of a **combinatorial optimization problem**. This is defined by a set of possible instances. An instance of MST is defined by a connected weighted graph. In general, an **instance** of a combinatorial optimization problem is described by a finite set $\mathcal{X}$ of allowed **configurations** and a **cost function** E defined on this set and taking values in $\mathbb{R}$. The optimization problem consists in finding the **optimal** configuration $C \in \mathcal{X}$, namely the one with the smallest cost $E(C)$. Any set of such instances defines a combinatorial optimization problem. For a particular instance of MST, the space of configurations $\mathcal{X}$ is simply the set of spanning trees on the given graph, while the cost function associated with each spanning tree is the sum of the costs of its edges.

We shall say that an algorithm solves an optimization problem if, for every instance of the optimization problem, it gives the optimal configuration, or if it computes its cost. In all the problems which we shall discuss, there is a 'natural' measure of the size of the problem N (typically a number of variables used to define a configuration, like the number of edges of the graph in MST), and the number of configurations scales, at large N like c^N, or in some cases even faster, e. g. like $N!$ or N^N. Notice that, quite generally, evaluating the cost function on a particular configuration is an easy task. The difficulty of solving the combinatorial optimization problem comes therefore essentially from the size of the configuration space.

It is a generally accepted practice to estimate the **complexity** of an algorithm as the number of 'elementary operations' required to solve the problem. Usually one focuses onto the asymptotic behavior of this quantity as $N \to \infty$. It is obviously of great practical interest to construct algorithms whose complexity is as small as possible.

One can solve a combinatorial optimization problem at several levels of refinement. Usually one distinguishes three types of problems:

- The **optimization** problem: Find an optimal configuration C^*.
- The **evaluation** problem: Determine the cost $E(C^*)$ of an optimal configuration.
- The **decision** problem: Answer to the question: "Is there a configuration of cost less than a given value E_0?"

3.3 More examples

The general setting described in the previous Section includes a large variety of problems having both practical and theoretical interest. In the following we shall provide a few selected examples.

3.3.1 Eulerian circuit

One of the oldest documented examples goes back to the 18th century. The old city of Königsberg had seven bridges (see Fig. 3.3), and its inhabitants were wondering whether it was possible to cross once each of this bridges and get back home. This can be generalized and translated in graph-theoretic language as the following decision problem. Define a **multigraph** exactly as a graph but for the fact that two given vertices can be connected by several edges. The problem consists in finding whether there is there a circuit which goes through all edges of the graph only once, and returns to its starting point. Such a circuit is now called a **Eulerian circuit**, because this problem was solved by Euler in 1736, when he proved the following nice theorem. As for ordinary graphs, we define the **degree** of a vertex as the number of edges which have the vertex as an end-point.

Theorem 3.2 *Given a connected multigraph, there exists an Eulerian circuit if and only if every vertex has even degree.*

This theorem directly provides an algorithm for the decision problem whose complexity grows linearly with the number of vertices of the graph: just go through all the vertices of the graph and check their degree.

Exercise 3.4 Show that, if an Eulerian circuit exists, the degrees are necessarily even.

Proving the inverse implication is more difficult. A possible approach consists in showing the following slightly stronger result. If all the vertices of a connected graph $\mathcal{G}$ have even degree but i and j, then there exists a path from i to j that visits once each edge in $\mathcal{G}$. This can be proved by induction on the number of vertices. [Hint: Start from i and make a step along the edge (i, i'). Show that it is possible to choose i' in such a way that the residual graph $\mathcal{G}\backslash(i, i')$ is connected.]

3.3.2 Hamiltonian cycle

More than a century after Euler's theorem, the great scientist sir William Hamilton introduced in 1859 a game called the icosian game. In its generalized form, it basically asks whether there exists, in a graph, a **Hamiltonian cycle**, that is a path going once through every vertex of the graph, and getting back to its starting point. This is another decision problem, and, at a first look, it seems very similar to the Eulerian circuit. However it turns out to be much more complicated. The best existing algorithms for determining the existence of an Hamiltonian cycle on a given graph run in a time which grows exponentially with the number of vertices N. Moreover, the theory of computational complexity, which we shall describe in Sec. 3.4, strongly suggests that this problem is in fact intrinsically difficult.

3.3.3 Traveling salesman

Given a complete graph with N points, and the distances d_{ij} between all pairs of points $1 \leq i < j \leq N$, the famous **traveling salesman problem** (TSP) is an optimization problem: find a Hamiltonian cycle of minimum total length. One can consider the case

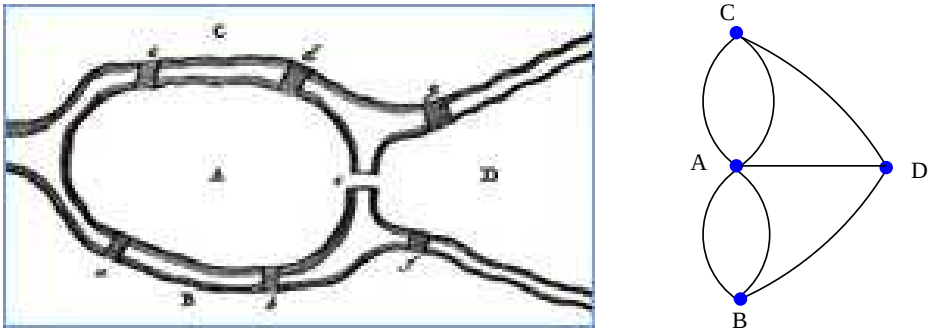

Fig. 3.3 Left: a map of the old city of Königsberg, with its seven bridges, as drawn in Euler's paper of 1736. The problem is whether one can walk along the city, crossing each bridge exactly once and getting back home. Right: a graph summarizing the problem. The vertices A, B, C, D are the various parts of lands separated by a river, an edge exists between two vertices whenever there is a bridge. The problem is to make a closed circuit on this graph, going exactly once through every edge.

where the points are in a portion of the plane, and the distances are Euclidean distances (we then speak of Euclidean TSP), but of course the problem can be stated more generally, with d_{ij} representing general costs, which are not necessarily distances. As for the Hamiltonian cycle problem, the best algorithms known so far for the TSP have a running time which grows exponentially with N at large N. Nevertheless Euclidean problems with thousands of points can be solved.

3.3.4 Assignment

Given N persons and N jobs, and a matrix C_{ij} giving the affinity of person i for job j, the **assignment** problem consists in finding the assignment of the jobs to the persons (an exact one-to-one correspondence between jobs and persons) which maximizes the total affinity. A configuration is characterized by a permutation of the N indices (there are thus $N!$ configurations), and the cost of the permutation π is $\sum_i C_{i\pi(i)}$. This is an example of a polynomial problem: there exists algorithms solving it in a time growing like N^3.

3.3.5 Satisfiability

In the **satisfiability** problem one has to find the values of N Boolean variables $x_i \in \{T, F\}$ which satisfy a set of logical constraints. Since each variable can be either true or false, the space of configurations has size $|\mathcal{X}| = 2^N$. Each logical constraint, called in this context a **clause**, takes a special form: it is the logical `OR` (for which we use the symbol $\vee$) of some variables or their negations. For instance $x_1 \vee \overline{x}_2$ is a 2-clause (a '2-clause' is a clause of length 2, i.e. which involves exactly 2 variables), which is satisfied if either $x_1 = T$, or $x_2 = F$, or both. Analogously $\overline{x}_1 \vee \overline{x}_2 \vee x_3$ is a 3-clause, which is satisfied by all configurations of the three variables except $x_1 = x_2 = T$, $x_3 = F$. The problem is to determine whether there exists a configuration which satisfies all constraints (decision problem), or to find the configuration which minimizes

the number of violated constraints (optimization problem). The K**-satisfiability** (or "K-SAT") problem is the restriction of satisfiability to the case where all clauses have length K. In 2-satisfiability the decision problem is easy: there exists an algorithm running in a time growing linearly with N. For K-satisfiability, and therefore also for the general satisfiability problem, all known algorithms solving the decision problem run in a time which grows exponentially with N.

3.3.6 Coloring and vertex covering

Given a graph and an integer q, the famous q**-coloring** problem asks if it is possible to color the vertices of the graph using q colors, in such a way that two vertices connected by an edge have different colors. In the same spirit, the **vertex-cover** problem asks to cover the vertices with 'pebbles', using the smallest possible number of pebbles, in such a way that every edge of the graph has at least one of its two endpoints covered by a pebble.

3.3.7 Number partitioning

Number partitioning is an example which does not come from graph theory. An instance is a set $\mathcal{S}$ of N integers $\mathcal{S} = \{x_1, .., x_N\}$. A configuration is a partition of these numbers into two groups $\mathcal{A}$ and $\mathcal{S} \setminus \mathcal{A}$. Is there a partition such that $\sum_{i\in\mathcal{A}} x_i = \sum_{i\in\mathcal{S}\setminus\mathcal{A}} x_i$?

3.4 Elements of the theory of computational complexity

One main branch of theoretical computer science aims at constructing an intrinsic theory of computational complexity. One would like, for instance, to establish which problems are harder than others. By 'harder problem', we mean a problem that takes a longer running time to be solved. In order to discuss rigorously the computational complexity of a problem, we would need to define a precise *model of computation* (introducing, for instance, Turing machines). This would take us too far. We will instead evaluate the running time of an algorithm in terms of 'elementary operations': comparisons, sums, multiplications, etc. This informal approach is essentially correct as long as the size of the operands remains uniformly bounded.

3.4.1 The worst case scenario

As we already mentioned in Sec. 3.2, a combinatorial optimization problem is defined by the set of its possible instances. Given an algorithm solving the problem, its running time will vary from instance to instance, even if the 'size' of the instance is fixed. How should we quantify the overall hardness of the problem? A crucial choice of computational complexity theory consists in considering the 'worst' (i.e. the one which takes longer time to be solved) instance among all the ones having the same size.

This choice has two advantages: (*i*) It allows to construct a 'universal' theory. (*ii*) Once the worst case running time of a given algorithm is estimated, this provides a performance guarantee on any instance of the problem.

3.4.2 Polynomial or not?

A second crucial choice consists in classifying algorithms in two classes: (*i*) **Polynomial**, if the running time is upper bounded by a fixed polynomial in the size of the instance. In mathematical terms, let T_N the number of operations required for solving an instance of size N in the worst case. The algorithm is polynomial when there exist a constant k such that $T_N = O(N^k)$. (*ii*) **Super-polynomial**, if no such upper bound exists. This is for instance the case if the time grows exponentially with the size of the instance (we shall call algorithms of this type **exponential**), i.e. $T_N = \Theta(k^N)$ for some constant $k > 1$.

Example 3.3 In 3.1.2, we were able to show that the running time of the MST algorithm is upper bounded by N^3, with N the number of vertices tin the graph. This implies that such an algorithm is polynomial.

Notice that we did not give a precise definition of the 'size' of a problem. One may wonder whether, changing the definition, a particular problem can be classified both as polynomial an as super-polynomial. Consider, for instance, the assignment problem with $2N$ points. One can define the size as being N, or $2N$, or even N^2 which is the number of possible person-job pairs. The last definition would be relevant if one would count, for instance, the number of entries in the person-job cost matrix. However, any of these 'natural' definitions of size are a polynomial function one of the other. Therefore they do not affect the classification of an algorithm as polynomial or super-polynomial. We will discard other definitions (such as e^N or $N!$) as 'unnatural', without any further ado. The reader can convince herself on each of the examples of the previous Section.

3.4.3 Optimization, evaluation, decision

In order to get a feeling of their relative levels of difficulty, let us come back for a while to the three types of optimization problems defined in Sec. 3.2, and study which one is the hardest.

Clearly, if the cost of any configuration can be computed in polynomial time, the evaluation problem is not harder than the optimization problem: if one can find the optimal configuration in polynomial time, one can compute its cost also in polynomial time. The decision problem (deciding whether there exists a configuration of cost smaller than a given E_0) is not harder than the evaluation problem. So the order of increasing difficulty is: decision, evaluation, optimization.

However, in many cases where the costs take discrete values, the evaluation problem is not harder than the decision problem, in the following sense. Suppose that we have a polynomial algorithm solving the decision problem, and that the costs of all configurations can be scaled to be integers in an interval $[0, E_{\max}]$ of length $E_{\max} = \exp\{O(N^k)\}$ for some $k > 0$. An algorithm solving the decision problem can be used to solve the evaluation problem by dichotomy: one first takes $E_0 = E_{\max}/2$. If there exists a configuration of energy smaller than E_0, one iterates with E_0 the center of the interval $[0, E_{\max}/2]$. In the opposite case, one iterates with E_0 the center of the interval

$[E_{\max}/2, E_{\max}]$. Clearly this procedure finds the cost of the optimal configuration(s) in a time which is also polynomial.

3.4.4 Polynomial reduction

One would like to compare the levels of difficulty of various *decision problems*. The notion of polynomial reduction formalizes the sentence "not harder than" which we used so far, and helps to get a classification of decision problems.

Roughly speaking, we say that a problem $\mathcal{B}$ is not harder than $\mathcal{A}$ if any efficient algorithm for $\mathcal{A}$ (if such an algorithm existed) could be used as a subroutine of an algorithm solving efficiently $\mathcal{B}$. More precisely, given two decision problems $\mathcal{A}$ and $\mathcal{B}$, one says that $\mathcal{B}$ is **polynomially reducible** to $\mathcal{A}$ if the following conditions hold:

1. There exists a mapping R which transforms any instance $\mathtt{I}$ of problem $\mathcal{B}$ into an instance $R(\mathtt{I})$ of problem $\mathcal{A}$, such that the solution (yes/no) of the instance $R(\mathtt{I})$ of $\mathcal{A}$ gives the solution (yes/no) of the instance $\mathtt{I}$ of $\mathcal{B}$.

2. The mapping $\mathtt{I} \mapsto R(\mathtt{I})$ can be computed in a time which is polynomial in the size of $\mathtt{I}$.

3. The size of $R(\mathtt{I})$ is polynomial in the size of $\mathtt{I}$. This is in fact a consequence of the previous assumptions but there is no harm in stating it explicitly.

A mapping R satisfying the above requirements is called a polynomial reduction. Constructing a polynomial reduction among two problems is an important achievement since it effectively reduces their study to the study of just one of them. Suppose for instance to have a polynomial algorithm $\mathrm{Alg}_{\mathcal{A}}$ for solving $\mathcal{A}$. Then a polynomial reduction of $\mathcal{B}$ to $\mathcal{A}$ can be used for constructing a polynomial algorithm for solving $\mathcal{B}$. Given an instance $\mathtt{I}$ of $\mathcal{B}$, the algorithm just compute $R(\mathtt{I})$, feeds it into the $\mathrm{Alg}_{\mathcal{A}}$, and outputs the output of $\mathrm{Alg}_{\mathcal{A}}$. Since the size of $R(\mathtt{I})$ is polynomial in the size of $\mathtt{I}$, the resulting algorithm for $\mathcal{B}$ is still polynomial.

Let us work out an explicit example. We will show that the problem of existence of a Hamiltonian cycle in a graph is polynomially reducible to the satisfiability problem.

Example 3.4 An instance of the Hamiltonian cycle problem is a graph with N vertices, labeled by $i \in \{1, ..., N\}$. If there exists a Hamiltonian cycle in the graph, it can be characterized by N^2 Boolean variables $x_{ri} \in \{0, 1\}$, where $x_{ri} = 1$ if vertex number i is the r'th vertex in the cycle, and $x_{ri} = 0$ otherwise (one can take for instance $x_{11} = 1$). We shall now write a number of constraints that the variables x_{ri} must satisfy in order for a Hamiltonian cycle to exist, and we shall ensure that these constraints take the forms of the clauses used in the satisfiability problem (identifying $x = 1$ as true, $x = 0$ as false):

- Each vertex $i \in \{1, \ldots, N\}$ must belong to the cycle: this can be written as the clause $x_{1i} \vee x_{2i} \vee \cdots \vee x_{Ni}$, which is satisfied only if at least one of the numbers $x_{1i}, x_{2i}, \ldots, x_{Ni}$ equals one.
- For every $r \in \{1, \ldots, N\}$, one vertex must be the r'th visited vertex in the cycle: $x_{r1} \vee x_{r2} \vee \cdots \vee x_{rN}$.
- Each vertex $i \in \{1, \ldots, N\}$ must be visited only once. This can be implemented through the $N(N-1)/2$ clauses $\bar{x}_{rj} \vee \bar{x}_{sj}$, for $1 \leq r < s \leq N$.
- For every $r \in \{1, \ldots, N\}$, there must be only one r'th visited vertex in the cycle. This can be implemented through the $N(N-1)/2$ clauses $\overline{x}_{ri} \vee \overline{x}_{rj}$, for $1 \leq i < j \leq N$.
- If two vertices $i < j$ which are not connected by an edge of the graph, these vertices should not appear consecutively in the list of vertices of the cycle. Therefore we add, for every such pair and for every $r \in \{1, \ldots, N\}$, the clauses $\overline{x}_{ri} \vee \overline{x}_{(r+1)j}$ and $\overline{x}_{rj} \vee \overline{x}_{(r+1)i}$ (with the 'cyclic' convention $N + 1 = 1$).

It is straightforward to show that the size of the satisfiability problem constructed in this way is polynomial in the size of the Hamiltonian cycle problem. We leave as an exercise to show that the set of all above clauses is a sufficient set: if the N^2 variables satisfy all the above constraints, they describe a Hamiltonian cycle.

3.4.5 Complexity classes

Let us continue to focus onto decision problems. The classification of these problems with respect to polynomiality is as follows:

- **Class P:** These are the **polynomial** problems, that can be solved by an algorithm running in polynomial time. An example, cf. Sec. 3.1, is the decision version of the minimum spanning tree (which asks for a yes/no answer to the question: given a graph with costs on the edges, and a number E_0, is there a spanning tree with total cost less than E_0?).
- **Class NP:** This is the class of **non-deterministic polynomial** problems, which can be solved in polynomial time by a 'non deterministic' algorithm. Roughly speaking, such an algorithm can run in parallel on an arbitrarily large number of processors. We shall not explain this notion in detail here, but rather use an alternative and equivalent characterization. We say that a problem is in the class NP if there exists a 'short' certificate which allows to check a 'yes' answer to the problem. A short certificate means a certificate that can be checked in polynomial time.

A polynomial problem like the MST described above is automatically in NP so P $\subseteq$ NP. The decision version of the TSP is also in NP: if there is a TSP tour with cost smaller than E_0, the short certificate is simple: just give the tour, and its cost will be computed in linear time, allowing to check that it is smaller than E_0. Satisfiability also belongs to NP: a certificate is obtained from the assignment of variables satisfying all clauses. Checking that all clauses are satisfied is linear in the number of clauses, taken here as the size of the system. In fact there are many important problems in the class NP, with a broad spectrum of applications ranging from routing to scheduling, to chip verification, or to protein folding...

- **Class NP-complete:** These are the hardest problem in the NP class. A problem is **NP-complete** if: (*i*) it is in NP, (*ii*) any other problem in NP can be polynomially reduced to it, using the notion of polynomial reduction defined in Sec. 3.4.4. If $\mathcal{A}$ is NP-complete, then: for any other problem $\mathcal{B}$ in NP, there is a polynomial reduction mapping $\mathcal{B}$ to $\mathcal{A}$. In other words, if we had a polynomial algorithm to solve $\mathcal{A}$, then all the problems in the broad class NP would be solved in polynomial time.

It is not *a priori* obvious whether there exist any NP-complete problem. A major achievement of the theory of computational complexity is the following theorem, obtained by Cook in 1971.

Theorem 3.5 *The satisfiability problem is NP-complete*

We shall not give here the proof of the theorem. Let us just mention that the satisfiability problem has a very universal structure (an example of which was shown above, in the polynomial reduction of the Hamiltonian cycle problem to satisfiability). A clause is built as the logical `OR` (denoted by $\vee$) of some variables, or their negations. A set of several clauses, to be satisfied simultaneously, is the logical `AND` (denoted by $\wedge$) of the clauses. Therefore a satisfiability problem is written in general in the form $(a_1 \vee a_2 \vee \dots) \wedge (b_1 \vee b_2 \vee \dots) \wedge \dots$, where the a_i, b_i are 'literals', i.e. any of the original variables or their negations. This form is called a **conjunctive normal form** (CNF), and it is easy to see that any logical statement between Boolean variables can be written as a CNF. This universal decomposition gives an idea of why the satisfiability problem plays a central role.

3.4.6 P=NP ?

When a NP-complete problem $\mathcal{A}$ is known, one can relatively easily find other NP-complete problems: if there exists a polynomial reduction from $\mathcal{A}$ to another problem $\mathcal{B} \in$ NP, then $\mathcal{B}$ is also NP-complete. In fact, whenever $R_{\mathcal{A}\leftarrow\mathcal{P}}$ is a polynomial reduction from a problem $\mathcal{P}$ to $\mathcal{A}$ and $R_{\mathcal{B}\leftarrow\mathcal{A}}$ is a polynomial reduction from $\mathcal{A}$ to $\mathcal{B}$, then $R_{\mathcal{B}\leftarrow\mathcal{A}} \circ R_{\mathcal{A}\leftarrow\mathcal{P}}$ is a polynomial reduction from $\mathcal{P}$ to $\mathcal{B}$. Starting from satisfiability, it has been possible to find, with this method, thousands of NP-complete problems. To quote a few of them, among the problems we have encountered so far, Hamiltonian circuit, TSP, and 3-satisfiability are NP-complete. Actually most of NP problems can be classified either as being in P, or being NP-complete. The precise status of some NP problems, like graph isomorphism, is still unknown.

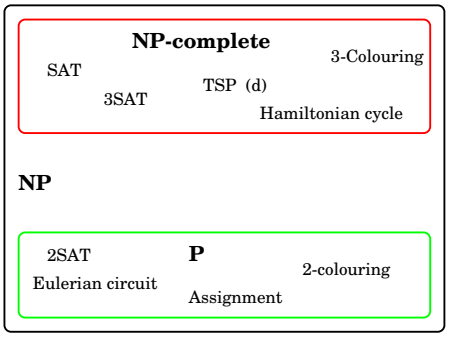

Fig. 3.4 Classification of some famous decision problems. If P $\neq$ NP, the classes P and NP-complete are disjoint. If it happened that $P = NP$, all the problems in NP, and in particular all those mentioned here, would be solvable in polynomial time.

Exercise 3.5 Show that 3-satisfiability is NP-complete, by constructing a polynomial reduction from satisfiability. The idea is to transform all possible clauses into sets of 3-clauses, using the following steps:

- A 2-clause $x_1 \vee x_2$ can be written as two 3-clauses $(x_1 \vee x_2 \vee y) \wedge (x_1 \vee x_2 \vee \overline{y})$ with one extra variable y.
- Write a 1-clause with four 3-clauses and two extra variables.
- Show that a k-clause $x_1 \vee x_2 \vee \cdots \vee x_k$ with $k \geq 4$ can be written with $k-3$ auxiliary variables as $(x_1 \vee x_2 \vee y_1) \wedge (x_3 \vee \overline{y}_1 \vee y_2) \wedge \cdots \wedge (x_{k-2} \vee \overline{y}_{k-4} \vee y_{k-3}) \wedge (x_{k-1} \vee x_k \vee \overline{y}_{k-3})$

Finally, those problems which, not being in NP are at least as hard as NP-complete problems, are usually called **NP-hard**. These includes both decision problems for which a short certificate does not exist, and non-decision problems. For instance the optimization and evaluation versions of TSP are NP-hard. However, in such cases, we shall chose among the expressions 'TSP is NP-complete' or 'TSP is NP-hard' rather freely.

One major open problem in the theory of computational complexity is whether the classes P and NP are distinct or not. It might be that P=NP=NP-complete: this would be the case if someone found a polynomial algorithm for one NP-complete problem. This would imply that any problem in the broad NP-class could be solved in polynomial time.

It is a widespread conjecture that there exists no polynomial algorithm for NP-complete problems. Then the classes P and NP-complete would be disjoint. Moreover it is known that, if P $\neq$ NP, then there are NP problems which are neither in P nor in NP-complete.

3.4.7 Other complexity classes

Notice the fundamental asymmetry in the definition of the NP class: the existence of a short certificate is requested only for the yes answers. To understand the meaning of this asymmetry, consider the problem of unsatisfiability (which is the complement of the satisfiability problem) formulated as: "given a set of clauses, is the problem unsatisfiable?". It is not clear if there exists a short certificate allowing to check a yes answer: it is very difficult to prove that a problem cannot be satisfied without checking an exponentially large number of possible configurations. So it is not at all obvious that unsatisfiability is in NP. Problems which are complements of those in NP define the class of co-NP problems, and it is not known whether NP=co-NP or not, although it is widely believed that co-NP is different from NP. This consideration opens a Pandora box with many other classes of complexities, but we shall immediately close it since it would carry us too far.

3.5 Optimization and statistical physics

3.5.1 General relation

There exists a natural mapping from optimization to statistical physics. Consider an optimization problem defined by a finite set $\mathcal{X}$ of allowed configurations, and a cost function E defined on this set with values in $\mathbb{R}$. While optimization consists in finding the configuration $C \in \mathcal{X}$ with the smallest cost, one can introduce a probability measure of the Boltzmann type on the space of configurations: For any β, each C is assigned a probability

$$\mu_\beta(C) = \frac{1}{Z(\beta)} e^{-\beta E(C)} \quad ; \quad Z(\beta) = \sum_{C \in \mathcal{X}} e^{-\beta E(C)} \ . \tag{3.1}$$

The positive parameter β plays the role of an inverse temperature. In the limit $\beta \to \infty$, the probability distribution μ_β concentrates on the configurations of minimum energy (ground states in the statistical physics jargon). This is the relevant limit for optimization problems. Notice that there exist many alternatives to the straightforward generalization (3.1). In some problems it may be useful to use more than one inverse temperature parameters β. Some of these parameters can be used to 'soften' constraints. For instance in the TSP one might like to relax the constraint that a configuration is a tour, by summing over all length N paths of the salesman, with an extra cost each time the path does not make a full tour, associated with inverse temperature β_1. The length of the path is another cost, associated with inverse temperature β_2. The original problem is recovered when both β_1 and β_2 go to infinity.

In the statistical physics approach one generalizes the optimization problem to study properties of the distribution μ_β at finite β. In many cases it is useful to follow μ_β when β increases (for instance by monitoring the thermodynamic properties: internal energy, entropy, the specific heat, ...). This may be particularly useful, both for analytical and for algorithmic purpose, when the thermodynamic properties evolve smoothly. An example of practical application is the simulated annealing method, which actually samples the configuration space at larger and larger values of β until it

finds a ground state. It will be described in Chap. **??**. As we will see, the occurrence of phase transitions poses major challenges to this kind of approaches.

3.5.2 Spin glasses and maximum cuts

To give a concrete example, let us go back to the spin glass problem of Sec. 2.6. This involves N Ising spins $\sigma_1, \ldots, \sigma_N$ in $\{\pm 1\}$, located on the vertices of a graph, and the energy function is:

$$E(\underline{\sigma}) = -\sum_{(ij)} J_{ij}\sigma_i\sigma_j, \tag{3.2}$$

where the sum $\sum_{(ij)}$ runs over all edges of the graph and the J_{ij} variables are exchange couplings which can be either positive or negative. Given the graph and the exchange couplings, what is the ground state of the corresponding spin glass? This is a typical optimization problem. In fact, it very well known in computer science in a slightly different form.

Each spin configuration partitions the set of vertices into two complementary subsets: $V_\pm = \{i \,|\, \sigma_i = \pm 1\}$. Let us call $\gamma(V_+)$ the set of edges with one endpoint in V_+, the other in V_-. The energy of the configuration can be written as:

$$E(\underline{\sigma}) = -C + 2\sum_{(ij)\in\gamma(V_+)} J_{ij}, \tag{3.3}$$

where $C = \sum_{(ij)} J_{ij}$. Finding the ground state of the spin glass is thus equivalent to finding a partition of the vertices, $V = V_+ \cup V_-$, such that $\sum_{(ij)\in\gamma(V_+)} c_{ij}$ is maximum, where $c_{ij} \equiv -J_{ij}$. This problem is known as the **maximum cut** problem (MAX-CUT): the set of edges $\gamma(V_+)$ is a cut, each cut is assigned a weight $\sum_{(ij)\in\gamma(V_+)} c_{ij}$, and one seeks the cut with maximal weight.

Standard results on max-cut immediately apply: In general this is an NP-hard problem, but there are some categories of graphs for which it is polynomially solvable. In particular the max-cut of a planar graph can be found in polynomial time, providing an efficient method to obtain the ground state of a spin glass on a square lattice in two dimensions. The three dimensional spin glass problem falls into the general NP-hard class, but efficient 'branch and bound' methods, based on its max-cut formulation, have been developed for this problem in recent years.

Another well known application of optimization to physics is the random field Ising model, which is a system of Ising spins with ferromagnetic couplings (all J_{ij} are positive), but with a magnetic field h_i which varies from site to site taking positive and negative values. Its ground state can be found in polynomial time thanks to the equivalence with the problem of finding a maximal flow in a graph.

3.6 Optimization and coding

Computational complexity issues are also crucial in all problems of information theory. We will see it recurrently throughout book, but let us just give here some small examples in order to fix ideas.

Consider the error correcting code problem of Chapter 1. We have a code, which maps an original message to a codeword $\underline{x}$, which is a point in the N-dimensional hypercube $\{0, 1\}^N$. There are 2^M codewords (with $M < N$), which we assume to be *a priori* equiprobable. When the message is transmitted through a noisy channel, the codeword $\underline{x}$ is corrupted to -say- a vector $\underline{y}$ with probability $Q(\underline{y}|\underline{x})$. The decoder maps the received message $\underline{y}$ to one of the possible input codewords $\underline{x}' = d(\underline{y})$.

As we saw, a measure of performance is the average block error probability:

$$\mathrm{P}_{\mathrm{B}}^{\mathrm{av}} \equiv \frac{1}{2^M} \sum_{\underline{x}} \sum_{\underline{y}} Q(\underline{y}|\underline{x}) \;\; \mathbb{I}(d(\underline{y}) \neq \underline{x}) \tag{3.4}$$

A simple decoding algorithm would be the following: for each received message $\underline{y}$, consider all the 2^M codewords, and determine the most likely one: $d(\underline{y}) = \arg\max_{\underline{x} \in \mathrm{Code}} Q(\underline{y}|\underline{x})$. It is clear that this algorithm minimizes the average block error probability.

For a general code, there is no better way for maximizing $Q(\underline{y}|\underline{x})$ than going through all codewords and computing their likelihood one by one. This takes a time of order 2^M, which is definitely too large. Recall in fact that, to achieve reliable communication, M and N have to be large (in data transmission application one may use N as large as 10^5). One might object that 'decoding a general code' is too a general problem. Just for specifying a single instance we would need to specify all the codewords, which takes $N\, 2^M$ bits. Therefore, the complexity of decoding could be a trivial consequence of the fact that even reading the input takes a huge time. However, it can be proved that, also for codes admitting a concise (polynomial in the blocklength) specification, decoding is NP-hard. We will see some examples, linear codes, in the following chapters.

Notes

We have left aside most algorithmic issues in this chapter. A general approach for designing efficient algorithms consists in finding a good 'convex relaxation' of the problem. The idea is to enlarge the space of feasible solution in such a way that the problem translates into minimizing a convex function, a task that can be performed efficiently. A general introduction to combinatorial optimization, including all these aspects, is provided by (Papadimitriou and Steiglitz, 1998). Convex optimization is the topic of many textbooks, for instance (Boyd and Vandenberghe, 2004).

The MST algorithm described in Sec. 3.1 was found in (Prim, 1957).

A complete treatment of computational complexity theory can be found in (Garey and Johnson, 1979), or in the more recent (Papadimitriou, 1994). The seminal theorem by Cook, (Cook, 1971), was independently rediscovered by Levin in 1973. The reader can find its proof in one of the above books.

Euler discussed the Könisberg's 7 bridges problem in (Euler, 1736).

The TSP, which is simple to state, difficult to solve, and lends itself to nice figures representations, has attracted lots of works. The interested reader can find many references, pictures of TSP's optimal tours with thousands of vertices, including tours among the main cities in various countries, applets, etc.. on the web, starting from instance from (Applegate, Bixby, Chvátal and Cook,).

The book (Hartmann and Rieger, 2002) focuses on the use of optimization algorithms for solving some problems in statistical physics. In particular it explains the

determination of the ground state of a random field Ising model with a maximum flow algorithm. A recent volume edited by these same authors (Hartmann and Rieger, 2004) addresses several algorithmic issues connecting optimization and physics. Chapter 4 by Liers, Jünger, Reinelt and Rinaldi describes the branch-and-cut approach to the maximum cut problem used for spin glass studies. The book (Hartmann and Weigt, 2005) contains an introduction to combinatorial optimization as a physics problem, with particular emphasis on the vertex cover problem.

Standard computational problems from coding theory are reviewed in (Barg, 1998). Some more recent issues are addressed by (Spielman, 1997). Finally, the first proof of NP-completeness for a decoding problem was obtained by (Berlekamp, McEliecee and van Tilborg, 1978).

5
The Random Energy Model

The random energy model (REM) is probably the simplest statistical physics model of a disordered system which exhibits a phase transition. It is not supposed to give a realistic description of any physical system, but it provides an example on which various concepts and methods can be studied in full detail. Moreover, due to its simplicity, the same mathematical structure appears in a large number of contexts. This is witnessed by the examples from information theory and combinatorial optimization presented in the next two chapters. The model is defined in Sec. 5.1 and its thermodynamic properties are studied in Sec. 5.2. The simple approach developed in these section turns out to be useful in a large variety of problems. A more detailed, and more involved, study of the low temperature phase is developed in Sec. 5.3. Section 5.4 provides an introduction to the so-called annealed approximation, which will be useful in more complicated models. Finally, in Sec. 5.5 we consider a variation of the REM that which is a cartoon for the structure of the set of solutions of random constraint satisfaction problems.

5.1 Definition of the model

A statistical mechanics model is defined by a set of configurations and an energy function defined on this space. In the REM there are $M = 2^N$ configurations (like in a system of N Ising spins) to be denoted by indices $i, j, \cdots \in \{1, \dots, 2^N\}$. The REM is a **disordered model**: the energy is not a deterministic function but rather a stochastic process. A particular realization of such a process is usually called a **sample** (or **instance**). In the REM, one makes the simplest possible choice for this process: the energies $\{E_i\}$ are i.i.d. random variables (the energy of a configuration is also called an **energy level**). For definiteness we shall keep here to the case where they have Gaussian distribution with zero mean and variance $N/2$, but other distributions could be studied as well. The scaling with N of the distribution should always be chosen in such a way that thermodynamic potentials are extensive. The pdf for the energy E_i of the state i is given by

$$P(E) = \frac{1}{\sqrt{\pi N}}\, e^{-E^2/N} \ , \tag{5.1}$$

Given an instance of the REM, defined by the 2^N real numbers $\{E_1, E_2, \dots, E_{2^N}\}$, one assigns to each configuration j a Boltzmann probability $\mu_\beta(j)$ in the usual way:

$$\mu_\beta(j) = \frac{1}{Z} \exp\left(-\beta E_j\right) \tag{5.2}$$

where $\beta = 1/T$ is the inverse of the temperature, and the normalization factor Z, the partition function, equals:

$$Z = \sum_{j=1}^{2^N} \exp\left(-\beta E_j\right) . \tag{5.3}$$

Notice that Z depends upon the temperature β, the 'sample size' N, and the particular realization of the energy levels $E_1, \ldots, E_M$. We shall write $Z = Z_N(\beta)$ to emphasize the dependency of the partition function upon N and β.

It is important not to be confused by the existence of two levels of probabilities in the REM, as in all disordered systems. We are interested in the properties of a probability distribution, the Boltzmann distribution (5.2), which is itself a random object because the energy levels are random variables.

Physically, a particular realization of the energy function corresponds to a given sample of some substance whose microscopic features cannot be controlled experimentally. This is what happens, for instance, in a metallic alloy: only the proportions of the various components can be controlled. The precise positions of the atoms of each species are described as random variables. The expectation value with respect to the sample realization will be denoted in the following by $\mathbb{E}(\cdot)$. For a given sample, Boltzmann's law (5.2) gives the probability of occupying the various possible configurations, according to their energies. The average with respect to Boltzmann distribution will be denoted by $\langle \cdot \rangle$. In experiments one deals with a single (or a few) sample(s) of a given disordered material. One could therefore be interested in computing the various thermodynamic potential (free energy F_N, internal energy U_N , or entropy S_N) for *this given* sample. This is an extremely difficult task. However, in most cases, as $N \to \infty$, the probability distributions of intensive thermodynamic potentials concentrate around their expected values. Formally, for any tolerance $\theta > 0$

$$\lim_{N\to\infty} \mathbb{P}\left[\left|\frac{X_N}{N} - \mathbb{E}\left(\frac{X_N}{N}\right)\right| \geq \theta\right] = 0 \tag{5.4}$$

where X is a thermodynamic potential ($X = F, S, U, \ldots$). In statistical phisics, the quantity X is then said to be **self-averaging** (in probability theory, one says that it **concentrates**). This essential property can be summarized plainly by saying that almost all large samples "behave" in the same way. This is the reason why different samples of alloys with the same chemical composition have the same thermodynamic properties. Often the convergence is exponentially fast in N (this happens for instance in the REM): this means that the expected value $\mathbb{E}\, X_N$ provides a good description of the system already at moderate sizes.

5.2 Thermodynamics of the REM

In this Section we compute the thermodynamic potentials of the REM in the thermodynamic limit $N \to \infty$. Our strategy consists in estimating the microcanonical entropy density, which has been introduced in Sec. 2.4. This knowledge is then used for computing the partition function Z to leading exponential order at large N.

5.2.1 Direct evaluation of the entropy

Let us consider an interval of energies $\mathcal{I} = [N\varepsilon, N(\varepsilon+\delta)]$, and call $\mathcal{N}(\varepsilon, \varepsilon+\delta)$ the number of configurations i such that $E_i \in \mathcal{I}$. Each energy level E_i belongs to $\mathcal{I}$ independently with probability:

$$P_{\mathcal{I}} = \sqrt{\frac{N}{\pi}} \int_{\varepsilon}^{\varepsilon+\delta} e^{-Nx^2} \, \mathrm{d}x \, . \tag{5.5}$$

Therefore $\mathcal{N}(\varepsilon, \varepsilon+\delta)$ is a binomial random variable, and its expectation and variance are given by:

$$\mathbb{E}\mathcal{N}(\varepsilon, \varepsilon+\delta) = 2^N \, P_{\mathcal{I}}, \qquad \mathrm{Var}\,\mathcal{N}(\varepsilon, \varepsilon+\delta) = 2^N \, P_{\mathcal{I}}[1 - P_{\mathcal{I}}] \, , \tag{5.6}$$

Because of the appropriate scaling with N of the interval $\mathcal{I}$, the probability $P_{\mathcal{I}}$ depends exponentially upon N. To exponential accuracy we thus have

$$\mathbb{E}\mathcal{N}(\varepsilon, \varepsilon+\delta) \doteq \exp\left\{ N \max_{x\in[\varepsilon,\varepsilon+\delta]} s_{\mathrm{a}}(x) \right\} \, , \tag{5.7}$$

$$\frac{\mathrm{Var}\mathcal{N}(\varepsilon, \varepsilon+\delta)}{[\mathbb{E}\mathcal{N}(\varepsilon, \varepsilon+\delta)]^2} \doteq \exp\left\{ -N \max_{x\in[\varepsilon,\varepsilon+\delta]} s_{\mathrm{a}}(x) \right\} \tag{5.8}$$

where $s_{\mathrm{a}}(x) \equiv \log 2 - x^2$. Notice that $s_{\mathrm{a}}(x) \geq 0$ if and only if $x \in [-\varepsilon_*, \varepsilon_*]$, with $\varepsilon_* = \sqrt{\log 2}$.

The intuitive content of these equalities is the following: When ε is outside the interval $[-\varepsilon_*, \varepsilon_*]$, the typical density of energy levels is exponentially small in N: for a generic sample there is no configuration at energy $E_i \approx N\varepsilon$. On the contrary, when $\varepsilon \in]-\varepsilon_*, \varepsilon_*[$, there is an exponentially large density of levels, and the fluctuations of this density are very small. This result is illustrated by a small numerical experiment in Fig. 5.1. We now give a more formal version of this statement.

Proposition 5.1 *Define the entropy function*

$$s(\varepsilon) = \begin{cases} s_{\mathrm{a}}(\varepsilon) = \log 2 - \varepsilon^2 & \text{if } |\varepsilon| \leq \varepsilon_*, \\ -\infty & \text{if } |\varepsilon| > \varepsilon_*. \end{cases} \tag{5.9}$$

Then, for any couple ε and δ, with probability one:

$$\lim_{N\to\infty} \frac{1}{N} \log \mathcal{N}(\varepsilon, \varepsilon+\delta) = \sup_{x\in[\varepsilon,\varepsilon+\delta]} s(x) \ . \tag{5.10}$$

Proof: The proof makes a simple use of the two moments of the number of energy levels in $\mathcal{I}$, found in (5.7,5.8).

Let us first assume that the interval $[\varepsilon, \varepsilon+\delta]$ is disjoint from $[-\varepsilon_*, \varepsilon_*]$. Then $\mathbb{E}\mathcal{N}(\varepsilon, \varepsilon+\delta) \doteq e^{-AN}$, with $A = -\sup_{x\in[\varepsilon,\varepsilon+\delta]} s_{\mathrm{a}}(x) > 0$. As $\mathcal{N}(\varepsilon, \varepsilon+\delta)$ is an integer, we have the simple inequality

$$\mathbb{P}[\mathcal{N}(\varepsilon, \varepsilon+\delta) > 0] \leq \mathbb{E}\mathcal{N}(\varepsilon, \varepsilon+\delta) \doteq e^{-AN} \, . \tag{5.11}$$

In words, the probability of having an energy level in any fixed interval outside $[-\varepsilon_*, \varepsilon_*]$ is exponentially small in N. The inequality of the form (5.11) goes under the name of

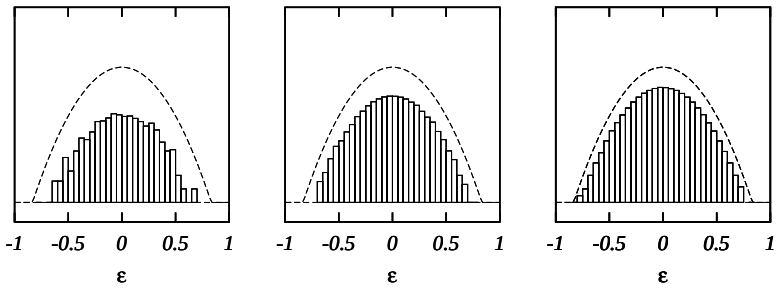

Fig. 5.1 Histogram of the energy levels for three samples of the random energy model with increasing sizes: from left to right $N = 10, 15$ and 20. Here we plot $N^{-1} \log \mathcal{N}(\varepsilon, \varepsilon + \delta)$ versus ε, with $\delta = 0.05$. The dashed curve gives the $N \to \infty$ analytical prediction (5.9).

Markov inequality, and the general strategy is sometimes called the **first moment method**.

Assume now that the intersection between $[\varepsilon, \varepsilon+\delta]$ and $[-\varepsilon_*, \varepsilon_*]$ is a finite length interval. In this case $\mathcal{N}(\varepsilon, \varepsilon+\delta)$ is tightly concentrated around its expectation $\mathbb{E}\mathcal{N}(\varepsilon, \varepsilon+\delta)$ as can be shown using Chebyshev inequality. For any fixed $C > 0$ one has

$$\mathbb{P}\left\{ \left| \frac{\mathcal{N}(\varepsilon, \varepsilon+\delta)}{\mathbb{E}\mathcal{N}(\varepsilon, \varepsilon+\delta)} - 1 \right| > C \right\} \leq \frac{\mathrm{Var}\, \mathcal{N}(\varepsilon, \varepsilon+\delta)^2}{C^2 [\mathbb{E}\mathcal{N}(\varepsilon, \varepsilon+\delta)]^2} \doteq e^{-BN} , \tag{5.12}$$

with $B = \sup_{x \in [\varepsilon, \varepsilon+\delta]} s_{\mathrm{a}}(x) > 0$.

Finally, the statement (5.10) follows from the previous estimates through a straightfoward application of the Borel-Cantelli Lemma. □

Exercise 5.1 (Large deviations.) Let $\mathcal{N}_{\mathrm{out}}(\delta)$ be the total number of configurations j such that $|E_j| > N(\varepsilon_* + \delta)$, with $\delta > 0$. Use Markov inequality to show that the fraction of samples in which there exist such configurations is exponentially small.

Besides being an interesting mathematical statement, Proposition 5.1 provides a good quantitative estimate. As shown in Fig. 5.1, already at $N = 20$, the outcome of a numerical experiment is quite close to the asymptotic prediction. Notice that, for energies in the interval $]-\varepsilon_*, \varepsilon_*[$, most of the discrepancy is due to the fact that we dropped subexponential factors in $\mathbb{E}\mathcal{N}(\varepsilon, \varepsilon + \delta)$: This produces corrections of order $\Theta(\log N/N)$ to the asymptotic behavior (5.10). The contribution due to fluctuations of $\mathcal{N}(\varepsilon, \varepsilon + \delta)$ around its average is instead exponentially small in N.

5.2.2 Thermodynamics and phase transition

From the previous result on the microcanonical entropy density, we now compute the partition function $Z_N(\beta) = \sum_{i=1}^{2^N} \exp(-\beta E_i)$. In particular, we are interested in intensive thermodynamic potentials like the free-entropy density $\phi(\beta) = \lim_{N\to\infty} [\log Z_N(\beta)]/N$. We start with a quick (and loose) argument, using the general approach outlined in

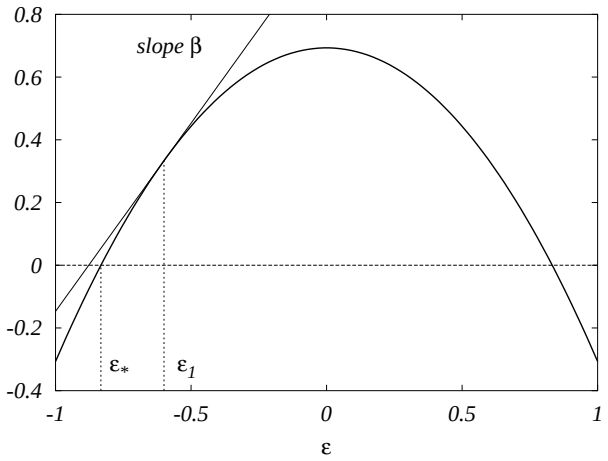

Fig. 5.2 The 'annealed' entropy density $s_{\rm a}(\varepsilon) = \log 2 - \varepsilon^2$ of the REM as a function of the energy density ε. The canonical entropy density $s(\beta)$ is the ordinate of the point with slope $\mathrm{d}s_{\rm a}/\mathrm{d}\varepsilon = \beta$ when this point lies within the interval $[-\varepsilon_*, \varepsilon_*]$ (this is for instance the case at $\varepsilon = \varepsilon_1$ in the plot), and $s(\beta) = 0$ otherwise. This gives rise to a phase transition at $\beta_{\rm c} = 2\sqrt{\log 2}$.

Sec. 2.4. It amounts to discretizing the energy axis using some step δ, and counting the energy levels in each interval with (5.10). Taking in the end the limit $\delta \to 0$ (after the limit $N \to \infty$), one expects to get, to leading exponential order:

$$Z_N(\beta) \doteq \int_{-\varepsilon_*}^{\varepsilon_*} \exp\left[N\left(s_{\rm a}(\varepsilon) - \beta\varepsilon\right)\right] \, \mathrm{d}\epsilon \; . \tag{5.13}$$

The rigorous formulation of the result can be obtained in analogy[1] with the general equivalence relation stated in Proposition 2.6. We find the free-entropy density:

$$\phi(\beta) = \max_{\varepsilon \in [-\varepsilon_*, \varepsilon_*]} \left[s_{\rm a}(\varepsilon) - \beta\varepsilon\right] , \tag{5.14}$$

Notice that although every sample of the REM is a new statistical physics system, with its own thermodynamic potentials, we have found that, with high probability, a random sample has free-entropy (or free-energy) density arbitrarily close to (5.14). A little more work shows that the internal energy and entropy density do concentrate as well. More precisely, for any fixed tolerance $\theta > 0$, we have $|(1/N)\log Z_N(\beta) - \phi(\beta)| < \theta$ with probability approaching one as $N \to \infty$.

Let us now discuss the physical content of the result (5.14). The optimization problem on the right-hand side can be solved through the geometrical construction illustrated in Fig. 5.2. One has to find a tangent to the curve $s_{\rm a}(\varepsilon) = \log 2 - \varepsilon^2$ with slope $\beta \geq 0$. Call $\varepsilon_{\rm a}(\beta) = -\beta/2$ the abscissa of the tangent point. If $\varepsilon_{\rm a}(\beta) \in [-\varepsilon_*, \varepsilon_*]$,

[1]The task is however more difficult here, because the density of energy levels $\mathcal{N}(\varepsilon, \varepsilon + \delta)$ is a random function whose fluctuations must be controlled.

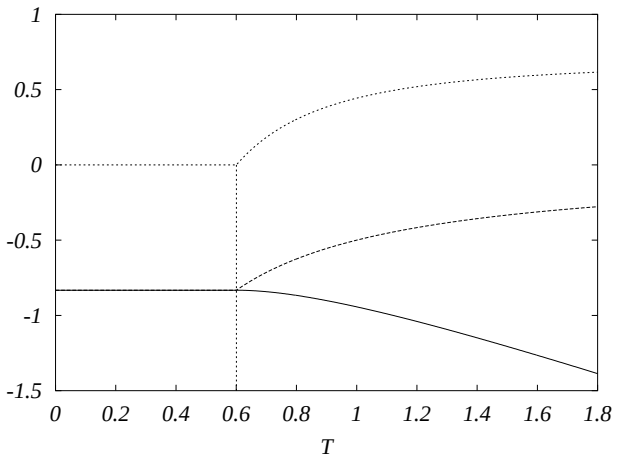

Fig. 5.3 Thermodynamics of the REM: the free-energy density (full line), the energy density (dashed line) and the entropy density (dotted line) are plotted versus temperature $T = 1/\beta$. The phase transition takes place at $T_c = 1/(2\sqrt{\log 2}) \approx 0.6005612$.

then the max in Eq. (5.14) is realized in $\varepsilon_a(\beta)$. In the other case $\varepsilon_a(\beta) < -\varepsilon_*$ (because $\beta \geq 0$) and the max is realized in $-\varepsilon_*$. Therefore:

Proposition 5.2 *The free-energy density of the REM, $f(\beta) = -\phi(\beta)/\beta$, is equal to:*

$$f(\beta) = \begin{cases} -\frac{1}{4}\beta - \log 2/\beta & \text{if } \beta \leq \beta_c\,, \\ -\sqrt{\log 2} & \text{if } \beta > \beta_c\,, \end{cases} \qquad \text{where} \quad \beta_c = 2\sqrt{\log 2}\;. \tag{5.15}$$

This shows that a phase transition (i.e. a non-analyticity of the free-energy density) takes place at the inverse critical temperature $\beta_c = 1/T_c = 2\sqrt{\log 2}$. It is a second order phase transition in the sense that the derivative of $f(\beta)$ is continuous, but because of the condensation phenomenon which we will discuss in Sec. 5.3 it is often called a 'random first order' transition. The other thermodynamic potentials are obtained through the usual formulas, cf. Sec. 2.2. They are plotted in Fig. 5.3.

The two temperature regimes -or 'phases'- , $\beta < \beta_c$ and $\beta > \beta_c$, have distinct qualitative properties which are most easily characterized through the thermodynamic potentials.

- In the high temperature phase $\beta \leq \beta_c$, the energy and entropy densities are: $u(\beta) = -\beta/2$ and $s(\beta) = \log 2 - \beta^2/4$. Boltzmann measure is dominated by configurations with energy $E_i \approx -N\beta/2$. There is an exponentially large number of configurations having such an energy density (the microcanonical entropy density $s(\varepsilon)$ is strictly positive at $\varepsilon = -\beta/2$), and the Boltzmann measure is roughly equidistributed among such configurations. In the infinite temperature limit $\beta \to 0$ it becomes uniform, and one finds as expected $u(\beta) \to 0$ (because nearly all configurations have an energy E_i/N close to 0) and $s \to \log 2$.

- In the low temperature phase $\beta > \beta_c$, the thermodynamic potentials are constant: $u(\beta) = -\varepsilon_*$ and $s(\beta) = 0$. The relevant configurations are the ones with the lowest energy density, namely with $E_i/N \approx -\varepsilon_*$. Boltzmann measure is dominated by a relatively small set of configurations, which is not exponentially large in N (the entropy density vanishes).

Exercise 5.2 The REM was originally motivated as a simple model of spin glass. One can generalize it by introducing the effect of a magnetic field B. The 2^N configurations are divided in $N+1$ groups. Each group is labelled by its 'magnetization' $M \in \{-N, -N+2, \dots, N-2, N\}$, and includes $\binom{N}{(N+M)/2}$ configurations. Their energies $\{E_j\}$ are independent Gaussian variables with variance $\sqrt{N/2}$ as in (5.1), and mean $\mathbb{E}\, E_j = -MB$ which depends upon the group j belongs to. Show that there exists a phase transition line $\beta_c(B)$ in the plane β, B such that:

$$\frac{1}{N}\mathbb{E}\, M = \begin{cases} \tanh[\beta B] & \text{when} \quad \beta \le \beta_c(B)\,, \\ \tanh[\beta_c(B) B] & \text{when} \quad \beta > \beta_c(B)\,, \end{cases} \tag{5.16}$$

and plot the magnetic susceptibility $\frac{\partial M}{\partial B}\big|_\beta = 0$ versus $T = 1/\beta$.

Exercise 5.3 Consider a generalization of the REM where the pdf of energies, instead of being Gaussian, is $P(E) \propto \exp\left[-C|E|^\delta\right]$, where $\delta > 0$. Show that, in order to have extensive thermodynamic potentials, one should scale C as $C = N^{1-\delta}\widehat{C}$ (i.e. the thermodynamic limit $N \to \infty$ should be taken at fixed $\widehat{C}$). Compute the critical temperature and the ground state energy density. What is the qualitative difference between the cases $\delta > 1$ and $\delta < 1$?

5.3 The condensation phenomenon

In the low temperature phase a smaller-than-exponential set of configurations dominates Boltzmann's measure: we say that a **condensation** of the measure onto these configurations takes place. This is a scenario which is typical of the appearance of a glass phase, and we shall encounter it in several other problems, including for instance satisfiability or colouring. It usually leads to many difficulties in finding the relevant configurations. In order to characterize the condensation, one can compute a **participation ratio** $Y_N(\beta)$ defined from Boltzmann's weights (5.2) as:

$$Y_N(\beta) \equiv \sum_{j=1}^{2^N} \mu_\beta(j)^2 \;=\; \left[\sum_j e^{-2\beta E_j}\right] \left[\sum_j e^{-\beta E_j}\right]^{-2} . \tag{5.17}$$

One can think of $1/Y_N(\beta)$ as giving some estimate of the 'effective' number of configurations which contribute to the measure. If the measure were equidistributed on r levels, one would have $Y_N(\beta) = 1/r$.

The participation ratio can be expressed as $Y_N(\beta) = Z_N(2\beta)/Z_N(\beta)^2$, where $Z_N(\beta)$ is the partition function at inverse temperature β. The analysis in the previous Section showed that $Z_N(\beta) \doteq \exp[N(\log 2 + \beta^2/4)]$ with very small fluctuations when $\beta < \beta_c$, while $Z_N(\beta) \doteq \exp[N\beta\sqrt{\log 2}]$ when $\beta > \beta_c$. This indicates that $Y_N(\beta)$ is exponentially small in N for almost all samples in the high temperature phase $\beta < \beta_c$, in agreement with the fact that the measure is not condensed at high temperatures. In the low temperature phase, on the contrary, we shall see that $Y_N(\beta)$ is finite and fluctuates from sample to sample.

The computation of $\mathbb{E}\,Y$ (we drop hereafter its arguments N and β) in the low temperature phase is slightly involved. It requires to control of the energy levels E_i with $E_i/N \approx -\varepsilon_*$. We give here a sketch of the computation, and leave the details to the reader as an exercise. Using the integral representation $1/Z^2 = \int_0^\infty t \exp(-tZ)\,\mathrm{d}$, one gets (denoting $M = 2^N$):

$$\mathbb{E}\,Y = M\,\mathbb{E}\int_0^\infty t\ \exp\left[-2\beta E_1\right]\ \exp\left[-t\sum_{i=1}^{M} e^{-\beta E_i}\right]\ \mathrm{d}t = \tag{5.18}$$

$$= M\int_0^\infty t\ a(t)\ [1-b(t)]^{M-1}\ \mathrm{d}t\ , \tag{5.19}$$

where

$$a(t) \equiv \int \exp\left[-2\beta E - te^{-\beta E}\right]\ \mathrm{d}P(E)\,, \tag{5.20}$$

$$b(t) \equiv \int [1-\exp(-te^{-\beta E})]\ \mathrm{d}P(E)\ , \tag{5.21}$$

and $P(E)$ is the Gaussian distribution (5.1). For large N the leading contributions to $\mathbb{E}\,Y$ come from the regions where E is close to $-N\varepsilon_0$ and $\log t$ is close to $-N\beta\varepsilon_0$, where

$$\varepsilon_0 = \varepsilon_* - \frac{1}{2\varepsilon_*}\log\sqrt{\pi N} \tag{5.22}$$

is fixed by the condition $2^N P(-N\varepsilon_0) = 1$. It can be thought as a refined estimate for the energy density of the lowest energy configuration.

We thus change variables from E, t to u, θ through $E = -N\varepsilon_0 + u$ and $t = \theta\exp(-N\beta\varepsilon_0)$, and we study the regime where u and θ are finite as $N \to \infty$. In this regime, the function $P(E)$ can be replaced by $2^{-N}e^{\beta_c u}$. One gets:

$$a(t) \simeq \frac{1}{M}\,e^{2N\beta\varepsilon_0}\int_{-\infty}^{+\infty} du\ \ e^{\beta_c u - 2\beta u - ze^{-\beta u}} = \frac{e^{2N\beta\varepsilon_0}}{M\beta}\ z^{\beta_c/\beta-2}\,\Gamma(2-\beta_c/\beta)\,, \tag{5.23}$$

$$b(t) \simeq \frac{1}{M}\int_{-\infty}^{+\infty} du\ \ e^{\beta_c u}\,[1-\exp(-ze^{-\beta u})] = -\frac{1}{M\beta}\ \ z^{\beta_c/\beta}\,\Gamma(-\beta_c/\beta)\,, \tag{5.24}$$

where $\Gamma(x)$ is Euler's Gamma function. Notice that the substitution $P(E) \simeq 2^{-N}e^{\beta_c u}$ is harmless because the resulting integrals (5.23) and (5.24) converge at large u.

At large N, the expression $[1-b(t)]^{M-1}$ in (5.19) can be approximated by $e^{-Mb(t)}$, and one finally obtains:

$$\mathbb{E}\,Y = M\int_0^\infty dt\ \ t\,a(t)\,e^{-Mb(t)}\ = \tag{5.25}$$
$$= \frac{1}{\beta}\Gamma\left(2-\frac{\beta_{\rm c}}{\beta}\right)\int_0^\infty dz\ \ z^{\beta_{\rm c}/\beta-1}\ \exp\left[\frac{1}{\beta}\Gamma\left(-\frac{\beta_{\rm c}}{\beta}\right)z^{\beta_{\rm c}/\beta}\right] = \ 1-\beta_{\rm c}/\beta\,,$$

where we used the approximate expressions (5.23), (5.24) and equalities are understood to hold up to corrections which vanish as $N\to\infty$.

We obtain therefore the following:

Proposition 5.3 *In the REM, when $N\to\infty$, the expectation value of the participation ratio is:*

$$\mathbb{E}\,Y = \begin{cases} 0 & \text{when } T > T_{\rm c}\,, \\ 1-T/T_{\rm c} & \text{when } T\le T_{\rm c}\,. \end{cases} \tag{5.26}$$

This gives a quantitative measure of the degree of condensation of Boltzmann's measure: when T decreases, the condensation starts at the phase transition temperature $T_{\rm c}$. At lower temperatures the participation ratio Y increases, meaning that the measure concentrates onto fewer and fewer configurations, until at $T=0$ only one configuration contributes and $Y=1$.

With the participation ratio we have a first qualitative and quantitative characterization of the low temperature phase. Actually the energies of the relevant configurations in this phase have many interesting probabilistic properties, to which we shall return in Chapter **??**.

5.4 A comment on quenched and annealed averages

In the previous section we have found that the self-averaging property holds in the REM, and this result allowed us to discuss the thermodynamics of a generic sample.

Self-averaging of the thermodynamic potentials is a very frequent property, but in more complicated systems it is often difficult to compute their expectation. We discuss here an approximation which is frequently used in such cases, the so-called annealed average. When the free-energy density is self averaging, the value of f_N is roughly the same for almost all samples and can be estimated by its expectation, called the **quenched average** $f_{N,{\rm q}}$:

$$f_{N,{\rm q}} = \mathbb{E}\,f_N = -\frac{T}{N}\mathbb{E}\log Z_N \tag{5.27}$$

This is the quantity that we computed in (5.15). In general it is hard to compute the expectation of the logarithm of the partition function. A much easier task is to compute the **annealed average**:

$$f_{N,{\rm a}} = -\frac{T}{N}\log(\mathbb{E}\,Z_N) \tag{5.28}$$

Let us compute it for the REM. Starting from the partition function (5.3), we find:

$$\mathbb{E}\, Z_N = \mathbb{E} \sum_{i=1}^{2^N} e^{-\beta E_i} = 2^N \mathbb{E}\, e^{-\beta E} = 2^N e^{N\beta^2/4} \,, \tag{5.29}$$

yielding $f_{N,\mathrm{a}}(\beta) = -\beta/4 - \log 2/\beta$.

Let us compare this with the correct free-energy density found in (5.15). Jensen's inequality (1.6) shows that the annealed free-energy density $f_{\mathrm{a}}(\beta)$ is always smaller than the correct one (remember that the logarithm is a concave function). In the REM, and a few other particularly simple problems, the annealed average gives the correct result in the high temperature phase $T > T_{\mathrm{c}}$, but fails to identify the phase transition, and predicts wrongly a free-energy density in the low temperature phase which is the analytic prolongation of the one at $T > T_{\mathrm{c}}$. In particular, it yields a *negative entropy density* $s_{\mathrm{a}}(\beta) = \log 2 - \beta^2/4$ for $T < T_{\mathrm{c}}$ (see Fig. 5.2).

A negative entropy is impossible in a system with finite configuration space, as can be seen from the definition of entropy. It thus signals a failure, and the reason is easily understood. For a given sample with free-energy density f, the partition function behaves as $Z_N = \exp(-\beta N f_N)$. Self-averaging means that f_N has small sample-to-sample fluctuations. However these fluctuations exist and are amplified in the partition function because of the factor N in the exponent. This implies that the annealed average of the partition function can be dominated by some very rare samples (those with an anomalously low value of f_N). Consider for instance the low temperature limit. We already know that in almost all samples the configuration with the lowest energy density is found at $E_i \approx -N\varepsilon_*$. However, there exist exceptional samples where one configuration has a smaller energy, $E_i = -N\varepsilon$, $\varepsilon > \varepsilon_*$. These samples are exponentially rare (they occur with probability $\doteq 2^N e^{-N\varepsilon^2}$), they are irrelevant as far as the quenched average is concerned, but they dominate the annealed average.

Let us add a short semantic note. The terms 'quenched' and 'annealed' originate in the thermal processing of materials used for instance in metallurgy of alloys: a quench corresponds to preparing a sample by bringing it suddenly from high to low temperatures. During a quench, atoms do not have time to change position (apart from some small vibrations). A given sample is formed by atoms at some random positions. On the contrary in an annealing process one gradually cools down the alloy, and the various atoms will find favorable positions. In the REM, the energy levels E_i are quenched: for each given sample, they take certain fixed values (like the positions of atoms in a quenched alloy). In the annealed approximation, one treats the configurations i and the energies E_i on the same footing. One says that the E_i variables are thermalized (like the positions of atoms in an annealed alloy).

In general, the annealed average can be used to find a lower bound on the free-energy in any system with finite configuration space. Useful results can be obtained for instance using the two simple relations, valid for all temperatures $T = 1/\beta$ and sizes N:

$$f_{N,\mathrm{q}}(T) \geq f_{N,\mathrm{a}}(T) \quad ; \quad \frac{df_{N,\mathrm{q}}(T)}{dT} \leq 0\,. \tag{5.30}$$

The first one follows from Jensen as mentioned above, while the second can be obtained from the positivity of canonical entropy, cf. Eq. (2.22), after averaging over the quenched disorder.

In particular, if one is interested in optimization problems (i.e. in the limit of vanishing temperature), the annealed average provides the general bound:

Proposition 5.4 *The ground state energy density*

$$u_N(T=0) \equiv \frac{1}{N}\mathbb{E}\left[\min_{\underline{x}\in\mathcal{X}^N} E(\underline{x})\right] . \tag{5.31}$$

satisfies the bound $u_N(0) \geq \max_{T\in[0,\infty]} f_{N,\mathrm{a}}(T)$

Proof: Consider the annealed free-energy density $f_{N,\mathrm{a}}(T)$ as a function of the temperature $T = 1/\beta$. For any given sample, the free-energy is a concave function of T because of the general relation (2.23). It is easy to show that the same property holds for the annealed average. Let T_* be the temperature at which $f_{N,\mathrm{a}}(T)$ achieves its maximum, and $f^*_{N,\mathrm{a}}$ be its maximum value. If $T_* = 0$, then $u_N(0) = f_{N,\mathrm{q}}(0) \geq f^*_{N,\mathrm{a}}$. It $T_* > 0$, using the two inequalities (5.30), one gets:

$$u_N(0) = f_{N,\mathrm{q}}(0) \geq f_{N,\mathrm{q}}(T_*) \geq f_{\mathrm{a}}(T_*) . \tag{5.32}$$

□

In the REM, this result immediately implies that $u(0) \geq \max_\beta[-\beta/4 - \log 2/\beta] = -\sqrt{\log 2}$, which is actually a tight bound.

5.5 The random subcube model

In the spirit of the REM, it is possible to construct a toy model for the set of solutions of a random constraint satisfaction problem. The **random subcube model** is defined by three parameters N, α, p. It has 2^N configurations: the vertices $\underline{x} = (x_1, \cdots, x_N)$ of the unit hypercube $\{0,1\}^N$. An instance of the model is defined by a subset $\mathcal{S}$ of the hypercube, the 'set of solutions.' Given an instance, the analogous of Boltzmann's measure is defined as the uniform distribution $\mu(\underline{x})$ over $\mathcal{S}$.

The solution space $\mathcal{S}$ is the union of $M = \lfloor 2^{(1-\alpha)N} \rfloor$ random subcubes which are i.i.d. subsets. Each subcube $\mathcal{C}_r$, $r \in \{1, \cdots, M\}$ is generated through the following procedure:

1. Generate the vector $t(r) = (t_1(r), t_2(r), \ldots, t_N(r))$, with independent entries

$$t_i(r) = \begin{cases} 0 \text{ with probability } (1-p)/2 , \\ 1 \text{ with probability } (1-p)/2 , \\ * \text{ with probability } p. \end{cases} \tag{5.33}$$

2. Given the values of $\{t_i(r)\}$, $\mathcal{C}_r$ is a subcube constructed as follows. For all i's such that $t_i(r)$ is 0 or 1, fix $x_i = t_i(r)$. Such variables are said to be 'frozen' for the subcube $\mathcal{C}_r$. For all other i's, x_i can be 0 or 1. These variables are said to be 'free'.

A configuration $\underline{x}$ may belong to several subcubes. Whenever it belongs to at least one subcube, it is in $\mathcal{S}$.

To summarize, $\alpha < 1$ fixes the number of subcubes, and $p \in [0,1]$ fixes their size. It can be studied using exactly the same methods as the REM. Here we shall just describe the main results, omitting all proofs. It is a *good exercise* to work out the details and prove the various assertions.

Let us denote by σ_r the entropy density of the r-th cluster in bits: $\sigma_r = (1/N)\log_2 |\mathcal{C}_r|$. It is clear that σ_r coincides with the fraction of $*$'s in the vector $t(r)$. In the large N limit, the number of clusters $\mathcal{N}(\sigma)$ with entropy density σ obeys a large deviation principle:

$$\mathcal{N}(\sigma) \doteq 2^{N\Sigma(\sigma)} \ . \tag{5.34}$$

The function $\Sigma(\sigma)$ is given as follows. Let $D(\sigma||p)$ denote the Kullback-Leibler distance between a Bernoulli σ and a Bernoulli p random variable. As we saw in Section 1.2, this is given by

$$D(\sigma||p) = \sigma \log_2 \frac{\sigma}{p} + (1-\sigma)\log_2 \frac{1-\sigma}{1-p} \ . \tag{5.35}$$

and define $[\sigma_1(p,\alpha), \sigma_2(p,\alpha)]$ as the interval in which $D(\sigma||p) \le 1-\alpha$. Then:

$$\Sigma(\sigma) = \begin{cases} 1-\alpha - D(\sigma||p) & \text{when } \sigma \in [\sigma_1(p,\alpha), \sigma_2(p,\alpha)] \ , \\ -\infty & \text{otherwise.} \end{cases} \tag{5.36}$$

We can now derive the phase diagram (see Fig. 5.4). Denote by s the total entropy density of the solution space, $s = (1/N)\log_2 |\mathcal{S}|$. Consider a configuration $\underline{x}$. The expected number of clusters to which it belongs is $2^{N(1-\alpha)}(\frac{1+p}{2})^N$. Therefore, if $\alpha < \alpha_{\rm d} \equiv \log_2(1+p)$, the solution space contains all but a vanishing fraction of the configurations, with high probability: $s = \log 2$. On the other hand, if $\alpha > \alpha_{\rm d}$, the probability that a configuration in $\mathcal{S}$ belongs to at least two distinct clusters is very small. In this regime $s = \max_\sigma(\Sigma(\sigma)+\sigma)$. As in the REM, there are two cases: (*i*) The maximum of $\Sigma(\sigma)+\sigma$ is achieved for $\sigma = \sigma_*(p,\alpha) \in]s_1(p,\alpha), s_2(p,\alpha)[$. This happens when $\alpha < \alpha_{\rm c}(p) \equiv \log_2(1+p) + (1-p)/(1+p)$. In this case $s = (1-\alpha)\log 2 + \log(1+p)$. (*ii*) The maximum of $\Sigma(\sigma)+\sigma$ is obtained for $\sigma = \sigma_2(p,\alpha)$. In this case $s = \sigma_2(p,\alpha)$.

Altogether we found three phases:

- For $\alpha < \alpha_{\rm d}$, subcubes overlap and one big cluster contains most configurations: $s_{\rm tot} = 1$
- For $\alpha_{\rm d} < \alpha < \alpha_{\rm c}$, the solution space $\mathcal{S}$ is splits into $2^{N(1-\alpha)}$ non overlapping clusters of configurations (every subcube is a cluster of solution, separated from the others). Most configurations of $\mathcal{S}$ are in the $e^{N\Sigma(s_*)}$ clusters which have entropy density close to $s_*(p,\alpha)$. Notice that the majority of clusters have entropy density $1-p < s_*$. There is a tension between the number of clusters and their size (i.e. their internal entropy). The result is that the less numerous, but larger, clusters with entropy density s_* dominate the uniform measure.
- For $\alpha > \alpha_{\rm c}$, the solution space $\mathcal{S}$ is still partitioned into $2^{N(1-\alpha)}$ non overlapping clusters of configurations. However most solutions are in clusters with entropy density close to $s_2(p,\alpha)$. The number of such clusters is not exponentially large. In fact the uniform measure over $\mathcal{S}$ shows a condensation phenomenon, which is completely analogous to the one of the REM. One can define a participation ratio $Y = \sum_r \mu(r)^2$, where $\mu(r)$ is the probability that a configuration of $\mathcal{S}$

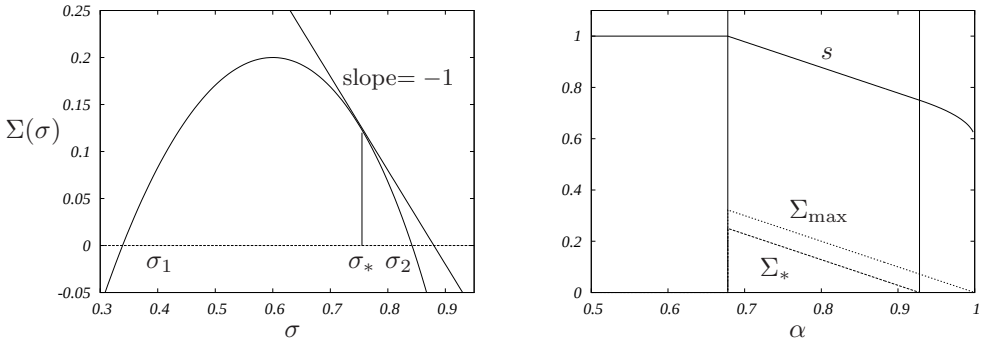

Fig. 5.4 Left: the function $\Sigma(\sigma)$ of the random subcubes model, for $p = 0.6$ and $\alpha = 0.8 \in]\alpha_{\rm d}, \alpha_{\rm c}[$. The maximum of the curve gives the total number of clusters $\Sigma_{\max}$. A 'typical' random solution $\underline{x} \in \mathcal{S}$ belongs to one of the $e^{N\Sigma(\sigma_*)}$ clusters with entropy density σ_*, with $\Sigma'(\sigma_*) = -1$. As α increases above $\alpha_{\rm c}$, random solutions condense in a few clusters with entropy density s_2. Right: thermodynamic quantities plotted versus α for $p = 0.6$: total entropy s, total number of clusters $\Sigma_{\max}$, and number of clusters where typical configurations are found Σ_*.

chosen uniformly at random belongs to cluster r, $\mu(r) = e^{N\sigma_r} / \sum_{r'} e^{N\sigma_{r'}}$. This participation ratio is finite, and equal to $1 - m$, where m is the slope $m = -\frac{{\rm d}\Sigma}{{\rm d}\sigma}$, evaluated at $s_2(p, \alpha)$.

Notes

The REM was invented in (Derrida, 1980), as an extreme case of some spin glass system. Here we have followed his original analysis which makes use of the microcanonical entropy. More detailed computations can be found in (Derrida, 1981) as well, including the solution to Exercise 2.

The condensation formula (5.3) appears first in (Gross and Mézard, 1984) as an application of replica computations which we shall discuss in Chapter **??**. The direct estimate of the participation ratio presented here and its fluctuations were developed in (Mézard, Parisi and Virasoro, 1985) and (Derrida and Toulouse, 1985). We shall return to the properties of the fascinating condensed phase in Chapter **??**.

Exercise 3 shows a phase transition which goes from second order for $\delta > 1$ to first order when $\delta < 1$. Its solution can be found in (Bouchaud and Mézard, 1997).

The random subcube model was introduced by (Achlioptas, 2007) and studied in detail in (Mora and Zdeborová, 2007). We refer to this paper for the derivations omitted from Sec. 5.5.

As a final remark, let us notice that in most of the physics literature, authors don't explicitely write down all the rigorous mathematical steps leading for instance to Eq. (5.13), preferring a more synthetic presentation which focuses on the basic ideas. In more complicated problems, it may be very difficult to fill the corresponding mathematical gaps. In many of the models studied in this book, it is still beyond the range of rigorous techniques. The recent book by Talagrand (Talagrand, 2003) adopts

a fully rigorous point of view, and it starts with a presentation of the REM which nicely complements the one given here and in Ch. **??**.

6
Random Code Ensemble

As already explained in Sec. 1.6, one of the basic problem of information theory consists in communicating reliably through a noisy communication channel. Error correcting codes achieve this task by systematically introducing some form of redundancy in the message to be transmitted. One of the major breakthrough accomplished by Claude Shannon was to understand the importance of code *ensembles*. He realized that it is much easier to construct ensembles of codes which have good properties with high probability, rather than exhibit explicit examples achieving the same performances. In a nutshell: 'stochastic' design is much easier than 'deterministic' design.

At the same time he defined and analyzed the simplest of such ensembles, which has been named thereafter the random code ensemble (or, sometimes, Shannon ensemble). Despite its great simplicity, the random code ensemble (RCE) has very interesting properties, and in particular it achieves optimal error correcting performances. It provides therefore a proof of the 'direct' part of the channel coding theorem: it is possible to communicate with vanishing error probability as long as the communication rate is smaller than the channel capacity. Furthermore, it is the prototype of a code based on a random construction. In the following Chapters we shall explore several examples of this approach, and the random code ensemble will serve as a reference.

We introduce the idea of code ensembles and define the RCE in 6.1. Some properties of this ensemble are described in Sec. 6.2, while its performances over the BSC are worked out in Sec. 6.3. We generalize these results to a general discrete memoryless channel in Sec. 6.4. Finally, in Sec. 6.5 we show that the RCE is optimal by a simple sphere-packing argument.

6.1 Code ensembles

An error correcting code is defined as a couple of encoding and decoding maps. The encoding map is applied to the information sequence to get an encoded message which is transmitted through the channel. The decoding map is applied to the (noisy) channel output. For the sake of simplicity, we shall assume throughout this Chapter that the message to be encoded is given as a sequence of M bits and that encoding produces a redundant sequence of $N > M$ bits. The possible codewords (i.e. the 2^M points in the space $\{0,1\}^N$ which are all the possible outputs of the encoding map) form the codebook $\mathfrak{C}_N$. On the other hand, we denote by $\mathcal{Y}$ the output alphabet of the communication channel. We use the notations

$$\underline{x} : \{0,1\}^M \to \{0,1\}^N \quad \text{encoding map}\,, \tag{6.1}$$

$$\underline{x}^{\mathrm{d}} : \quad \mathcal{Y}^N \to \{0,1\}^N \quad \text{decoding map}\,. \tag{6.2}$$

Notice that the definition of the decoding map is slightly different from the one given in Sec. 1.6. Here we consider only the difficult part of the decoding procedure, namely how to reconstruct from the received message the codeword which was sent. To complete the decoding as defined in Sec. 1.6, one should get back the original message knowing the codeword, but this is supposed to be an easier task (encoding is assumed to be injective).

The customary recipe for designing a **code ensemble** is the following: (*i*) Define a subset of the space of encoding maps (6.1); (*ii*) Endow this set with a probability distribution; (*iii*) Finally, for each encoding map in the ensemble, define the associated decoding map. In practice, this last step is accomplished by declaring that one among a few general 'decoding strategies' is adopted. We shall introduce a couple of such strategies below.

Our first example is the **random code ensemble (RCE)**. Notice that there exist 2^{N2^M} possible encoding maps of the type (6.1): one must specify N bits for each of the 2^M codewords. In the RCE, any of these encoding maps is picked with uniform probability. The code is therefore constructed as follows. For each of the possible information messages $m \in \{0,1\}^M$, we obtain the corresponding codeword $\underline{x}^{(m)} = (x_1^{(m)}, x_2^{(m)}, \dots, x_N^{(m)})$ by throwing N times an unbiased coin: the i-th outcome is assigned to the i-th coordinate $x_i^{(m)}$.

Exercise 6.1 Notice that, with this definition the code is not necessarily injective: there could be two information messages $m_1 \neq m_2$ with the same codeword: $\underline{x}^{(m_1)} = \underline{x}^{(m_2)}$. This is an annoying property for an error correcting code: each time that we send either of the messages m_1 or m_2, the receiver will not be able to distinguish between them, even in the absence of noise. Happily enough these unfortunate coincidences occur rarely, i.e. their number is much smaller than the total number of codewords 2^M. What is the expected number of couples m_1, m_2 such that $\underline{x}^{(m_1)} = \underline{x}^{(m_2)}$? What is the probability that all the codewords are distinct?

Let us now turn to the definition of the decoding map. We shall introduce here two among the most important decoding schemes: word MAP (MAP stands here for maximum *a posteriori* probability) and symbol MAP decoding. Both schemes can be applied to any code. In order to define them, it is useful to introduce the probability distribution $P(\underline{x}|\underline{y})$ for $\underline{x}$ to be the channel input conditional to the received message $\underline{y}$. For a memoryless channel with transition probability $Q(y|x)$, this probability has an explicit expression as a consequence of Bayes rule:

$$\mathbb{P}(\underline{x}|\underline{y}) = \frac{1}{Z(\underline{y})} \prod_{i=1}^{N} Q(y_i|x_i)\, \mathbb{P}(\underline{x})\,. \tag{6.3}$$

Here $Z(\underline{y})$ is fixed by the normalization condition $\sum_{\underline{x}} \mathbb{P}(\underline{x}|\underline{y}) = 1$, and $\mathbb{P}(\underline{x})$ is the *a priori* probability for $\underline{x}$ to be the transmitted message. Throughout this book, we shall assume that the transmitter chooses the codeword to be transmitted with uniform probability. Therefore $\mathbb{P}(\underline{x}) = 1/2^M$ if $\underline{x} \in \mathfrak{C}_N$ and $\mathbb{P}(\underline{x}) = 0$ otherwise. In formulas:

$$\mathbb{P}(\underline{x}) = \frac{1}{|\mathfrak{C}_N|}\, \mathbb{I}(\underline{x} \in \mathfrak{C}_N)\,. \tag{6.4}$$

Indeed it is not hard do realize that $Z(\underline{y})$ is the probability of observing $\underline{y}$ when a random codeword is transmitted. Hereafter we shall use $\mu_y(\,\cdot\,)$ to denote the *a posteriori* distribution (6.3) and $\mu_0(\,\cdot\,)$ for the *a priori* one (6.4). We can thus rewrite (6.3) as

$$\mu_y(\underline{x}) = \frac{1}{Z(\underline{y})} \prod_{i=1}^{N} Q(y_i|x_i)\, \mu_0(\underline{x})\,. \tag{6.5}$$

It is also useful to define the marginal distribution $\mu_y^{(i)}(x_i) = \mathbb{P}(x_i|\underline{y})$ of the i-th bit of the transmitted message conditional to the output message. This is obtained from the distribution (6.5) by marginalizing over all the bits x_j with $j \neq i$:

$$\mu_y^{(i)}(x_i) = \sum_{\underline{x}_{\setminus i}} \mu_y(\underline{x})\,, \tag{6.6}$$

where we introduced the shorthand $\underline{x}_{\setminus i} \equiv \{x_j : j \neq i\}$. **Word MAP** decoding outputs the most probable transmitted codeword, i.e. it maximizes the distribution (6.5)

$$\underline{x}^{\mathrm{w}}(\underline{y}) = \arg\max_{\underline{x}}\, \mu_y(\underline{x})\,. \tag{6.7}$$

We do not specify what to do in case of ties (i.e. if the maximum is degenerate), since this is irrelevant for all the coding problems that we shall consider. The scrupulous reader can chose her own convention in such cases.

A strongly related decoding strategy is **maximum-likelihood** decoding. In this case one maximizes $Q(\underline{y}|\underline{x})$ over $\underline{x} \in \mathfrak{C}_N$. This coincides with word MAP decoding whenever the *a priori* distribution over the transmitted codeword $\mathbb{P}(\underline{x}) = \mu_0(\underline{x})$ is taken to be uniform as in Eq. (6.4). Hereafter we will therefore blur the distinction between these two strategies.

Symbol (or bit) MAP decoding outputs the sequence of most probable transmitted bits, i.e. it maximizes the marginal distribution (6.6):

$$\underline{x}^{\mathrm{b}}(\underline{y}) = \left(\arg\max_{x_1} \mu_y^{(1)}(x_1)\,, \ldots, \arg\max_{x_N} \mu_y^{(N)}(x_N) \right)\,. \tag{6.8}$$

Exercise 6.2 Consider a code of block-length $N = 3$, and codebook size $|\mathfrak{C}| = 4$, with codewords $\underline{x}^{(1)} = \mathtt{001}$, $\underline{x}^{(1)} = \mathtt{101}$, $\underline{x}^{(1)} = \mathtt{110}$, $\underline{x}^{(1)} = \mathtt{111}$. What is the code rate? This code is used to communicate over a binary symmetric channel (BSC) with flip probability $p < 0.5$. Suppose that the channel output is $\underline{y} = \mathtt{000}$. Show that the word MAP decoding outputs the codeword $\mathtt{001}$. Now apply symbol MAP decoding to decode the first bit x_1: Show that the result coincides with the one of word MAP decoding only when p is small enough.

It is important to notice that each of the above decoding schemes is optimal with respect a different criterion. Word MAP decoding minimizes the average block error probability $\mathrm{P_B}$ defined in Sec. 1.6.2. This is the probability, with respect to the channel distribution $Q(\underline{y}|\underline{x})$, that the decoded codeword $\underline{x}^{\mathrm{d}}(\underline{y})$ is different from the transmitted one, averaged over the transmitted codeword:

$$\mathrm{P_B} \equiv \frac{1}{|\mathfrak{C}|} \sum_{\underline{x}\in\mathfrak{C}} \mathbb{P}[\underline{x}^{\mathrm{d}}(\underline{y}) \neq \underline{x}] \,. \tag{6.9}$$

Bit MAP decoding minimizes the **bit error probability**, or **bit error rate** (BER) $\mathrm{P_b}$. This is the fraction of incorrect bits, averaged over the transmitted codeword:

$$\mathrm{P_b} \equiv \frac{1}{|\mathfrak{C}|} \sum_{\underline{x}\in\mathfrak{C}} \frac{1}{N} \sum_{i=1}^{N} \mathbb{P}[x_i^{\mathrm{d}}(\underline{y}) \neq x_i] \,. \tag{6.10}$$

Exercise 6.3 Show that word MAP and symbol MAP decoding are indeed optimal with respect to the above criteria.

6.2 Geometry of the Random Code Ensemble

We begin our study of the RCE by first working out some of its geometrical properties. A code from this ensemble is defined by the codebook, a set $\mathfrak{C}_N$ of 2^M points (all the codewords) in the **Hamming space** $\{0,1\}^N$. Each of these points is drawn with uniform probability over the Hamming space. The simplest question one may ask about $\mathfrak{C}_N$ is the following. Suppose you sit on one of the codewords and look around you. How many other codewords are there at a given distance? We will use here the **Hamming distance**: the distance of two points $\underline{x}, \underline{y} \in \{0,1\}^N$ is the number of coordinates in which they differ.

This question is addressed through the **distance enumerator** $\mathcal{N}_{\underline{x}^{(0)}}(d)$ with respect to a codeword $\underline{x}^{(0)} \in \mathfrak{C}_N$. This is defined as the number of codewords in $\underline{x} \in \mathfrak{C}_N$ whose Hamming distance from $\underline{x}^{(0)}$ is equal to d: $d(\underline{x}, \underline{x}^{(0)}) = d$.

We shall now compute the typical properties of the distance enumerator for a random code. The simplest quantity to look at is the average distance enumerator $\mathbb{E}\mathcal{N}_{\underline{x}^{(0)}}(d)$, the average being taken over the code ensemble. In general one should further specify *which one* of the codewords is $\underline{x}^{(0)}$. Since in the RCE all codewords are drawn independently, and each one with uniform probability over the Hamming space, such a specification is irrelevant and we can in fact fix $\underline{x}^{(0)}$ to be the **all zeros codeword**, $\underline{x}^{(0)} = 000\cdots00$. Therefore we are asking the following question: take $2^M - 1$ point at random with uniform probability in the Hamming space $\{0,1\}^N$; what is the average number of points at distance d form the $00\cdots0$ corner? This is simply the number of points $(2^M - 1)$, times the fraction of the Hamming space 'volume' at a distance d from $000\cdots0$ $(2^{-N}\binom{N}{d})$:

$$\mathbb{E}\mathcal{N}_{\underline{x}^{(0)}}(d) = (2^M - 1)\, 2^{-N} \binom{N}{d} \quad \doteq \quad 2^{N[R-1+\mathcal{H}_2(\delta)]} \,. \tag{6.11}$$

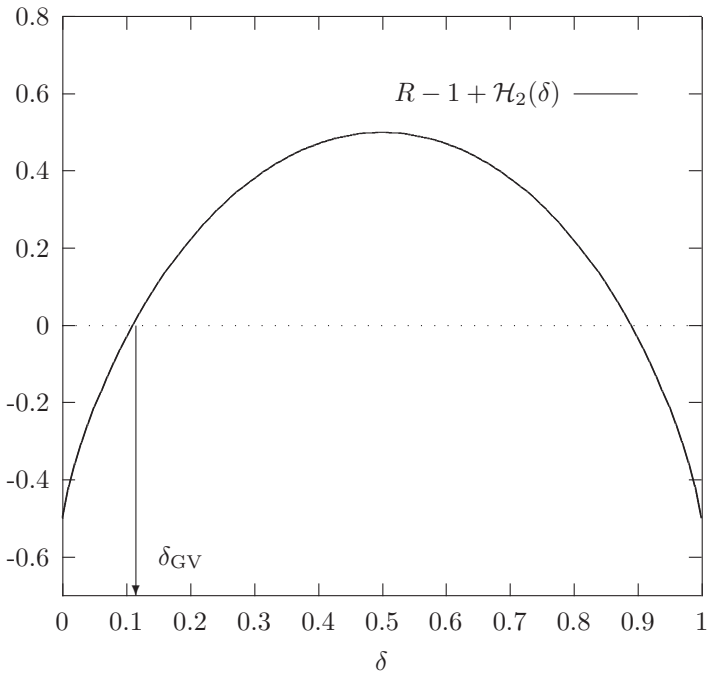

Fig. 6.1 Growth rate of the distance enumerator for the random code ensemble with rate $R = 1/2$ as a function of the Hamming distance $d = N\delta$.

In the second expression we introduced the fractional distance $\delta \equiv d/N$ and the rate $R \equiv M/N$, and considered the $N \to \infty$ asymptotics with these two quantities kept fixed. In Figure 6.1 we plot the function $R - 1 + \mathcal{H}_2(\delta)$ (which is sometimes called the **growth rate** of the distance enumerator). For δ small enough, $\delta < \delta_{GV}$, the growth rate is negative: the average number of codewords at small distance from $\underline{x}^{(0)}$ vanishes exponentially with N. By Markov inequality, the probability of having any codeword at all at such a short distance vanishes as $N \to \infty$. The distance $\delta_{\rm GV}(R)$, called the **Gilbert Varshamov distance**, is the smallest root of $R-1+\mathcal{H}_2(\delta) = 0$. For instance we have $\delta_{\rm GV}(1/2) \approx 0.110278644$.

Above the Gilbert Varshamov distance, $\delta > \delta_{\rm GV}$, the average number of codewords is exponentially large, with the maximum occurring at $\delta = 1/2$: $\mathbb{E}\mathcal{N}_{\underline{x}^{(0)}}(N/2) \doteq 2^{NR} = 2^M$. It is easy to show that the distance enumerator $\mathcal{N}_{\underline{x}^{(0)}}(d)$ is sharply concentrated around its average in this whole regime $\delta_{\rm GV} < \delta < 1 - \delta_{\rm GV}$. This is done using arguments similar to those developed in Sec.5.2 for the random energy model (REM configurations become codewords in the present context and the role of energy is played by Hamming distance; finally, the Gaussian distribution of the energy levels is replaced here by the binomial distribution). A pictorial interpretation of the above result is shown in Fig. 6.2 (notice that it is often misleading to interpret phenomena occurring in spaces with a large number of dimensions using finite dimensional images: such images must be handled with care!).

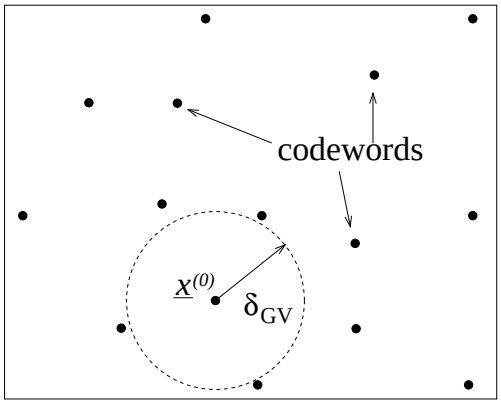

Fig. 6.2 A pictorial view of a typical code from the random code ensemble. The codewords are random points in the Hamming space. If we pick a codeword at random from the code and consider a ball of radius $N\delta$ around it, the ball will not contain any other codeword as long as $\delta < \delta_{\rm GV}(R)$, it will contain exponentially many codewords when $\delta > \delta_{\rm GV}(R)$

Exercise 6.4 The random code ensemble can be easily generalized to other (non binary) alphabets. Consider for instance a q-ary alphabet, i.e. an alphabet with letters $\{0, 1, 2, \ldots, q-1\} \equiv \mathcal{A}$. A code $\mathfrak{C}_N$ is constructed by taking 2^M codewords with uniform probability in $\mathcal{A}^N$. We can define the distance between any two codewords $d_q(\underline{x}, \underline{y})$ as the number of positions in which the sequence $\underline{x}$, $\underline{y}$ differ. Show that the average distance enumerator is now

$$\mathbb{E}\mathcal{N}_{\underline{x}^{(0)}}(d) \doteq 2^{N[R-\log_2 q+\delta\log_2(q-1)+\mathcal{H}_2(\delta)]}\,, \tag{6.12}$$

with $\delta \equiv d/N$ and $R \equiv M/N$. The maximum of the above function is no longer at $\delta = 1/2$. How can we explain this phenomenon in simple terms?

6.3 Communicating over the Binary Symmetric Channel

We shall now analyze the performances of the RCE when used for communicating over the binary symmetric channel (BSC) already defined in Fig. 1.4. We start by considering a word MAP (or, equivalently, maximum likelihood) decoder, and we analyze the slightly more complicated symbol MAP decoder afterwards. Finally, we introduce another decoding strategy inspired by the statistical physics analogy, that generalizes the word MAP and symbol MAP decoding.

6.3.1 Word MAP decoding

For a BSC, both the channel input $\underline{x}$ and output $\underline{y}$ are sequences of bits of length N. The probability for the codeword $\underline{x}$ to be the channel input conditional to the output

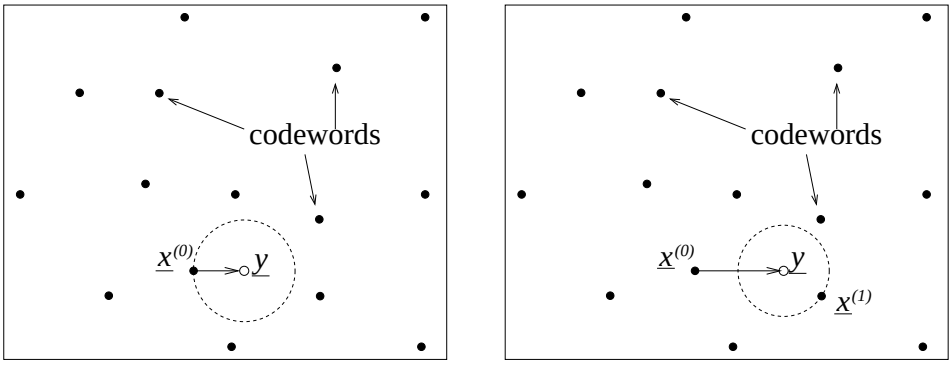

Fig. 6.3 A pictorial view of word MAP decoding for the BSC. A codeword $\underline{x}^{(0)}$ is chosen and transmitted through a noisy channel. The channel output is $\underline{y}$. If the distance between $\underline{x}^{(0)}$ and $\underline{y}$ is small enough (left frame), the transmitted message can be safely reconstructed by looking for the closest codeword to $\underline{y}$. In the opposite case (right frame), the closest codeword $\underline{x}_1$ does not coincide with the transmitted one.

$\underline{y}$, defined in Eqs. (6.5) and (6.4), depends uniquely on the Hamming distance $d(\underline{x},\underline{y})$ between these two vectors. Denoting by p the channel flip probability, we have

$$\mu_y(\underline{x}) = \frac{1}{Z'(\underline{y})}\, p^{d(\underline{x},\underline{y})}(1-p)^{N-d(\underline{x},\underline{y})}\, \mathbb{I}(\underline{x}\in\mathfrak{C}_N)\,, \tag{6.13}$$

$Z'(\underline{y})$ being a normalization constant which depends uniquely upon $\underline{y}$ (up to a factor, this coincides with the normalization $Z(\underline{y})$ in Eq. (6.5)). Without loss of generality, we can assume $p < 1/2$. Therefore word MAP decoding, which prescribes to maximize $\mu_y(\underline{x})$ with respect to $\underline{x}$, outputs the codeword which is the closest to the channel output.

We have obtained a purely geometrical formulation of the original communication problem. A random set of points $\mathfrak{C}_N$ is drawn in the Hamming space $\{0,1\}^N$ and one of them (let us call it $\underline{x}^{(0)}$) is chosen for communicating. The noise perturbs this vector yielding a new point $\underline{y}$. Decoding consists in finding the closest to $\underline{y}$ among all the points in $\mathfrak{C}_N$ and fails every time this is not $\underline{x}^{(0)}$. The block error probability is simply the probability for such an event to occur. This formulation is illustrated in Fig. 6.3.

This description should make immediately clear that the block error probability vanishes (in the $N\to\infty$ limit) as soon as p is below some finite threshold. In the previous Section we saw that, with high probability, the closest codeword $\underline{x}'\in\mathfrak{C}_N\backslash\underline{x}^{(0)}$ to $\underline{x}^{(0)}$ lies at a distance $d(\underline{x}',\underline{x}^{(0)})\simeq N\delta_{\rm GV}(R)$. On the other hand $\underline{y}$ is obtained from $\underline{x}^{(0)}$ by flipping each bit independently with probability p, therefore $d(\underline{y},\underline{x}^{(0)})\simeq Np$ with high probability. By the triangle inequality $\underline{x}^{(0)}$ is surely the closest codeword to $\underline{y}$ (and therefore word MAP decoding is successful) if $d(\underline{x}^{(0)},\underline{y}) < d(\underline{x}^{(0)},\underline{x}')/2$. If $p<\delta_{\rm GV}(R)/2$, this happens with probability approaching one as $N\to\infty$, and therefore the block error probability vanishes.

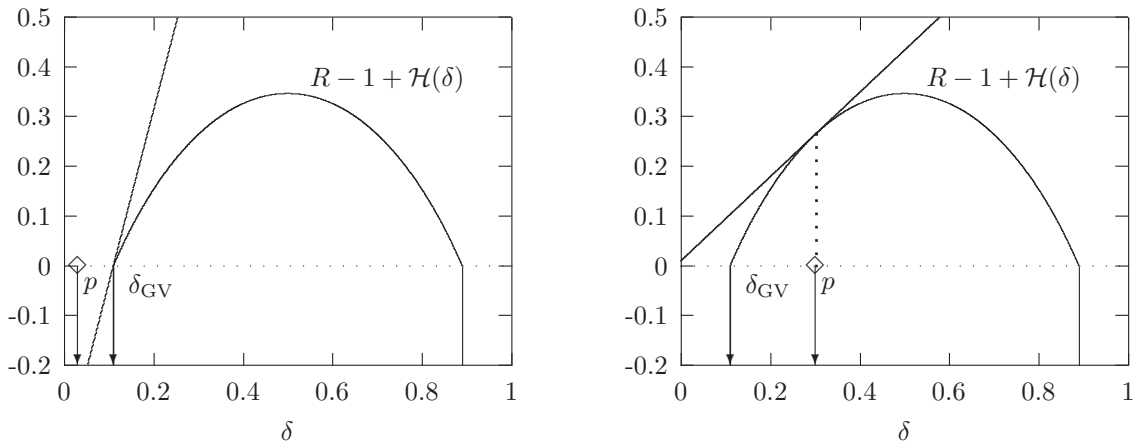

Fig. 6.4 Logarithm of the distance enumerator $\widehat{\mathcal{N}}_{\underline{y}}(d)$ (counting the number of codewords at a distance $d = N\delta$ from the received message) divided by the block-length N. Here the rate is $R = 1/2$. We also show the distance of the transmitted codeword for two different noise levels: $p = 0.03 < \delta_{\rm GV}(1/2) \approx 0.110278644$ (left) and $p = 0.3 > \delta_{\rm GV}(R)$ (right). The tangent lines with slope $2B = \log[(1-p)/p]$ determine which codewords dominate the symbol MAP decoder.

However the above argument overestimates the effect of noise. Although about $N\delta_{\rm GV}(R)/2$ incorrect bits may cause an unsuccessful decoding, they must occur in the appropriate positions for $\underline{y}$ to be closer to $\underline{x}'$ than to $\underline{x}^{(0)}$. If they occur at uniformly random positions (as it happens in the BSC) they will be probably harmless. The difference between the two situations is most significant in large-dimensional spaces, as shown by the analysis provided below.

The distance between $\underline{x}^{(0)}$ and $\underline{y}$ is the sum of N i.i.d. Bernoulli variables of parameter p (each bit gets flipped with probability p). By the central limit theorem, $N(p-\varepsilon) < d(\underline{x}^{(0)}, \underline{y}) < N(p+\varepsilon)$ with probability approaching one in the $N \to \infty$ limit, for any $\varepsilon > 0$. As for the remaining $2^M - 1$ codewords, they are completely uncorrelated with $\underline{x}^{(0)}$ and, therefore, with $\underline{y}$: $\{\underline{y}, \underline{x}^{(1)}, \cdots, \underline{x}^{(2^M-1)}\}$ are 2^M i.i.d. random points drawn from the uniform distribution over $\{0,1\}^N$. The analysis of the previous section shows that with probability approaching one as $N \to \infty$, none of the codewords $\{\underline{x}^{(1)}, \cdots, \underline{x}^{(2^M-1)}\}$ lies within a ball of radius $N\delta$ centered on $\underline{y}$, when $\delta < \delta_{\rm GV}(R)$. In the opposite case, if $\delta > \delta_{\rm GV}(R)$, there is an exponential (in N) number of these codewords within a ball of radius $N\delta$.

The performance of the RCE is easily deduced (see Fig. 6.4) : If $p < \delta_{\rm GV}(R)$, the transmitted codeword $\underline{x}^{(0)}$ lies at a shorter distance than all the other ones from the received message $\underline{y}$: decoding is successful. At a larger noise level, $p > \delta_{\rm GV}(R)$ there is an exponential number of codewords closer to $\underline{y}$ than the transmitted one: decoding is unsuccessful. Note that the condition $p < \delta_{\rm GV}(R)$ can be rewritten as $R < C_{\rm BSC}(p)$, where $C_{\rm BSC}(p) = 1 - \mathcal{H}_2(p)$ is the capacity of a BSC with flip probability p.

6.3.2 Symbol MAP decoding

In symbol MAP decoding, the i-th bit is decoded by first computing the marginal $P^{(i)}(x_i|\underline{y})$ and then maximizing it with respect to x_i. Using Eq. (6.13) we get

$$\mu_{\underline{y}}^{(i)}(x_i) = \sum_{\underline{x}_{\setminus i}} \mu_{\underline{y}}(\underline{x}) = \frac{1}{Z} \sum_{\underline{x}_{\setminus i}} \exp\{-2B\, d(\underline{x},\underline{y})\}\,, \tag{6.14}$$

where we introduced the parameter

$$B \equiv \frac{1}{2}\, \log\left(\frac{1-p}{p}\right)\,, \tag{6.15}$$

and the normalization constant

$$Z \equiv \sum_{\underline{x}\in\mathfrak{C}_N} \exp\{-2B\, d(\underline{x},\underline{y})\}\,. \tag{6.16}$$

Equation (6.14) shows that the marginal distribution $\mu_{\underline{y}}^{(i)}(x_i)$ sums contributions from all the codewords, not only from the one closest to $\underline{y}$. This makes the analysis of symbol MAP decoding slightly more involved than the word MAP decoding case.

Let us start by estimating the normalization constant Z. It is convenient to separate the contribution coming from the transmitted codeword $\underline{x}^{(0)}$ from the one of the *incorrect* codewords $\underline{x}^{(1)},\dots,\underline{x}^{(2^M-1)}$:

$$Z = e^{-2Bd(\underline{x}^{(0)},\underline{y})} + \sum_{d=0}^{N} \widehat{\mathcal{N}}_{\underline{y}}(d)\, e^{-2Bd} \equiv Z_{\rm corr} + Z_{\rm err}\,, \tag{6.17}$$

where we denoted by $\widehat{\mathcal{N}}_{\underline{y}}(d)$ the number of incorrect codewords at a distance d from the vector $\underline{y}$. The contribution of $\underline{x}^{(0)}$ in the above expression is easily estimated. By the law of large numbers $d(\underline{x}^{(0)},\underline{y}) \simeq Np$ and therefore $Z_{\rm corr}$ is close to e^{-2NBp} with high probability. More precisely, for any $\varepsilon > 0$, $e^{-N(2Bp+\varepsilon)} \le Z_{\rm corr} \le e^{-N(2Bp-\varepsilon)}$ with probability approaching one in the $N\to\infty$ limit.

As for $Z_{\rm err}$, one proceeds in two steps: first compute the distance enumerator $\widehat{\mathcal{N}}_{\underline{y}}(d)$, and then sum over d. The distance enumerator was already computed in Sec. 6.2. As in the word MAP decoding analysis, the fact that the distances are measured with respect to the channel output $\underline{y}$ and not with respect to a codeword does not change the result, because $\underline{y}$ is independent from the incorrect codewords $\underline{x}^{(1)}\cdots\underline{x}^{(2^M-1)}$. Therefore $\widehat{\mathcal{N}}_{\underline{y}}(d)$ is exponentially large in the interval $\delta_{\rm GV}(R) < \delta \equiv d/N < 1-\delta_{\rm GV}(R)$, while it vanishes with high probability outside the same interval. Moreover, if $\delta_{\rm GV}(R) < \delta < 1-\delta_{\rm GV}(R)$, $\widehat{\mathcal{N}}_{\underline{y}}(d)$ is tightly concentrated around its mean given by Eq. (6.11). The summation over d in Eq. (6.17) can then be evaluated by the saddle point method. This calculation is very similar to the estimation of the free-energy of the random energy model, cf. Sec. 5.2. Roughly speaking, we have

$$Z_{\rm err} = \sum_{d=0}^{N} \widehat{\mathcal{N}}_{\underline{y}}(d)\, e^{-2Bd} \simeq N \int_{\delta_{\rm GV}}^{1-\delta_{\rm GV}} e^{N[(R-1)\log 2 + \mathcal{H}(\delta) 2B\delta]}\, {\rm d}\delta \doteq e^{N\phi_{\rm err}}\,, \tag{6.18}$$

where

$$\phi_{\rm err} \equiv \max_{\delta\in[\delta_{\rm GV},1-\delta_{\rm GV}]} \big[\,(R-1)\log 2 + \mathcal{H}(\delta) - 2B\delta\,\big]\,. \tag{6.19}$$

The reader can complete the mathematical details of the above derivation as outlined in Sec. 5.2. The bottom line is that $Z_{\rm err}$ is close to $e^{N\phi_{\rm err}}$ with high probability as $N\to\infty$.

Let us examine the resulting expression (6.19) (see Fig. 6.4). If the maximum is achieved on the interior of $[\delta_{\rm GV}, 1-\delta_{\rm GV}]$, its location δ_* is determined by the stationarity condition $\mathcal{H}'(\delta_*) = 2B$, which implies $\delta_* = p$. In the opposite case, it must be realized at $\delta_* = \delta_{\rm GV}$ (remember that $B > 0$). Evaluating the right hand side of Eq. (6.19) in these two cases, we get

$$\phi_{\rm err} = \begin{cases} -\delta_{\rm GV}(R)\log\left(\frac{1-p}{p}\right) & \text{if } p < \delta_{\rm GV}, \\ (R-1)\log 2 - \log(1-p) & \text{otherwise.} \end{cases} \tag{6.20}$$

We can now compare $Z_{\rm corr}$ and $Z_{\rm err}$. At low noise level (small p), the transmitted codeword $\underline{x}^{(0)}$ is close enough to the received one $\underline{y}$ to dominate the sum in Eq. (6.17). At higher noise level, the exponentially more numerous incorrect codewords overcome the term due to $\underline{x}^{(0)}$. More precisely, with high probability we have

$$Z = \begin{cases} Z_{\rm corr}[1+e^{-\Theta(N)}] & \text{if } p < \delta_{\rm GV}, \\ Z_{\rm err}[1+e^{-\Theta(N)}] & \text{otherwise,} \end{cases} \tag{6.21}$$

where the $\Theta(N)$ exponents are understood to be positive.

We consider now Eq. (6.14), and once again separate the contribution of the transmitted codeword:

$$P^{(i)}(x_i|\underline{y}) = \frac{1}{Z}\left[Z_{\rm corr}\,\mathbb{I}(x_i = x_i^{(0)}) + Z_{{\rm err},x_i}\right], \tag{6.22}$$

where we have introduced the quantity

$$Z_{{\rm err},x_i} = \sum_{\underline{z}\in\mathfrak{C}_N\setminus\underline{x}^{(0)}} e^{-2Bd(\underline{z},\underline{y})}\,\mathbb{I}(z_i = x_i)\ . \tag{6.23}$$

Notice that $Z_{{\rm err},x_i} \le Z_{\rm err}$. Together with Eq. (6.21), this implies, if $p < \delta_{\rm GV}(R)$: $\mu_y^{(i)}(x_i = x_i^{(0)}) = 1 - e^{-\Theta(N)}$ and $\mu_y^{(i)}(x_i \neq x_i^{(0)}) = e^{-\Theta(N)}$. Therefore, in this regime, the symbol MAP decoder correctly outputs the transmitted bit $x_i^{(0)}$. It is important to stress that this result holds with probability approaching one as $N\to\infty$. Concretely, there exists bad choices of the code $\mathfrak{C}_N$ and particularly unfavorable channel realizations $\underline{y}$ such that $\mu_y^{(i)}(x_i = x_i^{(0)}) < 1/2$ and the decoder fails. However the probability of such an event (i.e. the bit-error rate $\rm P_b$) vanishes in the large blocklength limit $N\to\infty$.

What happens for $p > \delta_{\rm GV}(R)$? Arguing as for the normalization constant Z, it is easy to show that the contribution of incorrect codewords dominates the marginal

distribution (6.22). Intuitively, this suggests that the decoder fails. A more detailed computation, sketched below, shows that the bit error rate in the $N \to \infty$ limit is:

$$\mathrm{P_b} = \begin{cases} 0 & \text{if } p < \delta_{\mathrm{GV}}(R), \\ p & \text{if } \delta_{\mathrm{GV}}(R) < p < 1/2. \end{cases} \tag{6.24}$$

Notice that, above the threshold $\delta_{\mathrm{GV}}(R)$, the bit error rate is the same as if the information message were transmitted without coding through the BSC: the code is useless.

A complete calculation of the bit error rate $\mathrm{P_b}$ in the regime $p > \delta_{\mathrm{GV}}(R)$ is rather lengthy. We shall provide here an heuristic, albeit essentially correct, justification, and leave a more detailed derivation as the exercise below. As already stressed, the contribution Z_{corr} of the transmitted codeword can be safely neglected in Eq. (6.22). Assume, without loss of generality, that $x_i^{(0)} = \mathtt{0}$. The decoder will be successful if $Z_{\mathrm{err},\mathtt{0}} > Z_{\mathrm{err},\mathtt{1}}$ and fail in the opposite case. Two cases must be considered: either $y_i = \mathtt{0}$ (this happens with probability $1-p$), or $y_i = \mathtt{1}$ (probability p). In the first case we have

$$\begin{aligned} Z_{\mathrm{err},\mathtt{0}} &= \sum_{\underline{z} \in \mathfrak{C}_N \setminus \underline{x}^{(0)}} \mathbb{I}(z_i = \mathtt{0})\, e^{-2Bd_i(\underline{y},\underline{z})} \\ Z_{\mathrm{err},\mathtt{1}} &= e^{-2B} \sum_{\underline{z} \in \mathfrak{C}_N \setminus \underline{x}^{(0)}} \mathbb{I}(z_i = \mathtt{1})\, e^{-2Bd_i(\underline{y},\underline{z})}\,, \end{aligned} \tag{6.25}$$

where we denoted by $d_i(\underline{x},\underline{y})$ the number of of positions j, distinct form i, such that $x_j \neq y_j$. The sums in the above expressions are independent identically distributed random variables. Moreover they are tightly concentrated around their mean. Since $B > 0$, this implies $Z_{\mathrm{err},\mathtt{0}} > Z_{\mathrm{err},\mathtt{1}}$ with high probability. Therefore the decoder is successful in the case $y_i = \mathtt{0}$. Analogously, the decoder fails with high probability if $y_i = \mathtt{1}$, and hence the bit error rate converges to $\mathrm{P_b} = p$ for $p > \delta_{\mathrm{GV}}(R)$.

Exercise 6.5 From a rigorous point of view, the weak point of the above argument is the lack of any estimate of the fluctuations of $Z_{\mathrm{err},\mathtt{0}/\mathtt{1}}$. The reader may complete the derivation along the following lines:

(*a*) Define $X_{\mathtt{0}} \equiv Z_{\mathrm{err},\mathtt{0}}$ and $X_{\mathtt{1}} \equiv e^{2B}\, Z_{\mathrm{err},\mathtt{1}}$. Prove that $X_{\mathtt{0}}$ and $X_{\mathtt{1}}$ are independent and identically distributed.

(*b*) Define the correct distance enumerators $\mathcal{N}_{\mathtt{0}/\mathtt{1}}(d)$ such that a representation of the form $X_{\mathtt{0}/\mathtt{1}} = \sum_d \mathcal{N}_{\mathtt{0}/\mathtt{1}}(d)\, \exp(-2Bd)$ holds.

(*c*) Show that a significant fluctuation of $\mathcal{N}_{\mathtt{0}/\mathtt{1}}(d)$ from its average is highly (more than exponentially) improbable (within an appropriate range of d).

(*d*) Deduce that a significant fluctuation of $X_{\mathtt{0}/\mathtt{1}}$ is highly improbable (the last two points can be treated along the lines already discussed for the random energy model in Chapter 5).

6.3.3 Finite-temperature decoding

The expression (6.14) for the marginal $\mu_y^{(i)}(x_i)$ is strongly reminiscent of a Boltzmann average. This analogy suggests a generalization which interpolates between the two 'classical' MAP decoding strategies discussed so far: **finite-temperature decoding**. We first define this new decoding strategy in the context of the BSC context. Let β be a non-negative number playing the role of an inverse temperature, and $\underline{y} \in \{0,1\}^N$ the channel output. Define the probability distribution $\mu_{y,\beta}(\underline{x})$ to be given by

$$\mu_{y,\beta}(\underline{x}) = \frac{1}{Z(\beta)}\, e^{-2\beta B d(\underline{y},\underline{x})}\, \mathbb{I}(x \in \mathfrak{C}_N)\,, \qquad Z(\beta) \equiv \sum_{\underline{x}\in\mathfrak{C}_N} e^{-2\beta B d(\underline{x},\underline{y})}\,, \quad (6.26)$$

where B is always related to the noise level p through Eq. (6.15). This distribution depends upon the channel output $\underline{y}$: for each received message $\underline{y}$, the finite-temperature decoder constructs the appropriate distribution $\mu_{y,\beta}(\underline{x})$. For the sake of simplicity we don't write this dependence explicitly. Let $\mu_{y,\beta}^{(i)}(x_i)$ be the marginal distribution of x_i when $\underline{x}$ is distributed according to $\mu_{y,\beta}(\underline{x})$. The new decoder outputs

$$\underline{x}^{\beta} = \left(\arg\max_{x_1} \mu_{y,\beta}^{(1)}(x_1)\,, \dots, \arg\max_{x_N} \mu_{y,\beta}^{(N)}(x_N) \right)\,. \quad (6.27)$$

As in the previous Sections, the reader is free to choose her favorite convention in the case of ties (i.e. for those i's such that $\mu_{y,\beta}^{(i)}(0) = \mu_{y,\beta}^{(i)}(1)$).

Two values of β are particularly interesting: $\beta = 1$ and $\beta = \infty$. If $\beta = 1$ the distribution $\mu_{y,\beta}(\underline{x})$ coincides with the distribution $\mu_y(\underline{x})$ of the channel input conditional to the output, see Eq. (6.13). Therefore, for any $\underline{y}$, symbol MAP decoding coincides with finite-temperature decoding at $\beta = 1$: $\underline{x}_i^{\beta=1} = \underline{x}^{\rm b}$.

If $\beta = \infty$, the distribution (6.26) concentrates over those codewords which are the closest to $\underline{y}$. In particular, if there is a unique closest codeword to $\underline{y}$, finite-temperature decoding at $\beta = \infty$ coincides with word MAP decoding: $\underline{x}^{\beta=\infty} = \underline{x}^{\rm w}$.

Exercise 6.6 Using the approach developed in the previous Section, analyze the performances of finite-temperature decoding for the RCE at any β.

The results of the last exercise are summarized in Fig. 6.5 which give the finite-temperature decoding phase diagram. There exist three regimes which are three distinct phases with very different behaviors.

1. A 'completely ordered' phase at low noise ($p < \delta_{\rm GV}(R)$) and low temperature (large enough β). In this regime the decoder works: the probability distribution $\mu_{y,\beta}(\underline{x})$ is dominated by the transmitted codeword $\underline{x}^{(0)}$. More precisely $\mu_{y,\beta}(\underline{x}^{(0)}) = 1 - \exp\{-\Theta(N)\}$. The bit and block error rates vanish as $N \to \infty$.
2. A 'glassy' phase at higher noise ($p > \delta_{\rm GV}(R)$) and low temperature (large enough β). The transmitted codeword has a negligible weight $\mu_{y,\beta}(\underline{x}^{(0)}) = \exp\{-\Theta(N)\}$. The bit error rate is bounded away from 0, and the block error rate converges to

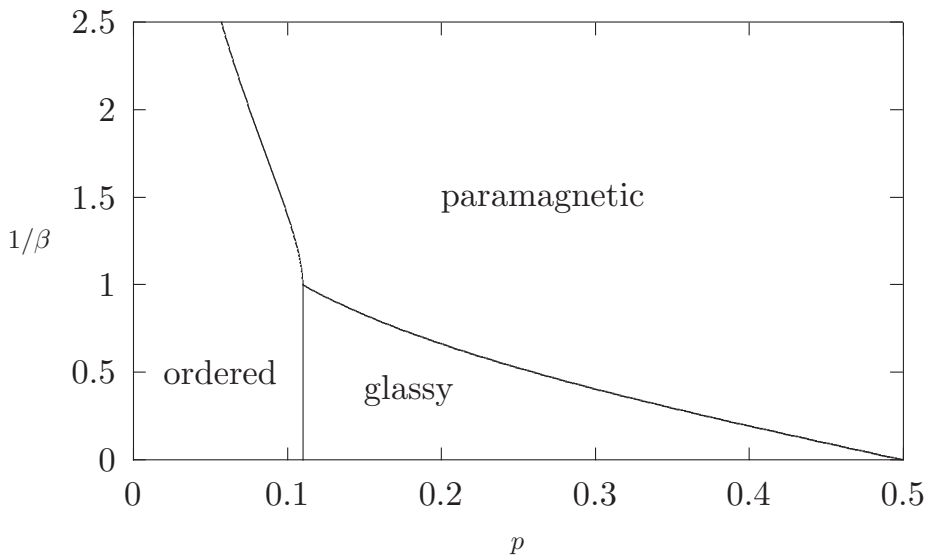

Fig. 6.5 Phase diagram for the rate 1/2 random code ensemble over the binary symmetric channel, using finite temperature decoding. Word MAP and bit MAP decoding correspond (respectively) to $1/\beta = 0$ and $1/\beta = 1$. Notice that the phase boundary of the error-free (ordered) phase is vertical in this interval of temperatures.

1 as $N \to \infty$. The measure $\mu_{y,\beta}(\underline{x})$ is dominated by the closest codewords to the received message $\underline{y}$ (which are distinct from the correct one since $p > \delta_{\rm GV}(R)$). Its Shannon entropy $H(\mu_{y,\beta})$ is sub-linear in N. This situation is closely related to the 'measure condensation' phenomenon occurring in the low-temperature phase of the random energy model.

3. An 'entropy dominated' (paramagnetic) phase at high temperature (small enough β). The bit and block error rates behave as in the glassy phase, and $\mu_{y,\beta}(\underline{x}^{(0)}) = \exp\{-\Theta(N)\}$. However the measure $\mu_{y,\beta}(\underline{x})$ is now dominated by codewords whose distance $d \simeq N\delta_*$ from the received message is larger than the minimal one: $\delta_* = p^\beta/[p^\beta + (1-p)^\beta]$. In particular $\delta_* = p$ if $\beta = 1$, and $\delta_* = 1/2$ if $\beta = 0$. In the first case we recover the result already obtained for symbol MAP decoding. In the second one, $\mu_{y,\beta=0}(\underline{x})$ is the uniform distribution over the codewords and the distance from the received message under this distribution is, with high probability, close to $N/2$. In this regime, the Shannon entropy $H(\mu_\beta)$ is linear in N.

Finite-temperature decoding can be generalized to other channel models. Let $\mu_y(\underline{x})$ be the distribution of the transmitted message conditional to the channel output, given explicitly in Eq. (6.5). For $\beta > 0$, we define the distribution[1]

[1]The partition function $Z(\beta)$ defined here differs by a multiplicative constant from the one defined in Eq. (6.26) for the BSC.

$$\mu_{y,\beta}(\underline{x}) = \frac{1}{Z(\beta)}\,\mu_y(\underline{x})^\beta\,, \qquad Z(\beta) \equiv \sum_{\underline{x}} \mu_y(\underline{x})^\beta\,. \tag{6.28}$$

Once more, the decoder decision for the i-th bit is taken according to the rule (6.27). The distribution $\mu_{y,\beta}(\underline{x})$ is a 'deformation' of the conditional distribution $\mu_y(\underline{x})$. At large β, more weight is given to highly probable transmitted messages. At small β the most numerous codewords dominate the sum. A little thought shows that, as for the BSC, the cases $\beta = 1$ and $\beta = \infty$ correspond, respectively, to symbol MAP and word MAP decoding. The qualitative features of the finite-temperature decoding phase diagram are easily generalized to any memoryless channel. In particular, the three phases described above can be found in such a general context. Decoding is successful in the low noise, large β phase.

6.4 Error-free communication with random codes

As we have seen, the block error rate $\mathrm{P_B}$ for communicating over a BSC with a random code and word MAP decoding vanishes in the large blocklength limit as long as $R < C_{\rm BSC}(p)$, with $C_{\rm BSC}(p) = 1 - \mathcal{H}_2(p)$ the channel capacity. This establishes the 'direct' part of Shannon's channel coding theoremfor the BSC case: error-free communication is possible at rates below the channel capacity. This result is in fact much more general. We describe here a proof for general memoryless channels, always based on random codes.

For the sake of simplicity we shall restrict ourselves to memoryless channels with binary input and discrete output. These are defined by a transition probability $Q(y|x)$, $x \in \{0,1\}$ and $y \in \mathcal{Y}$ with $\mathcal{Y}$ a finite alphabet. In order to handle this case, we must generalize the RCE: each codeword $\underline{x}^{(m)} \in \{0,1\}^N$, $m = 0, \dots, 2^M - 1$, is again constructed independently as a sequence of N i.i.d. bits $x_1^{(m)} \cdots x_N^{(m)}$. Unlike for symmetric channels, $x_i^{(m)}$ is now drawn from an arbitrary distribution $P(x)$, $x \in \{0,1\}$ instead of being uniformly distributed. It is important to distinguish $P(x)$, which is an arbitrary single bit distribution defining the code ensemble and will be chosen at our convenience for optimizing it, from the *a priori* source distribution $\mu_0(\underline{x})$ of Eq. (6.5), which is a distribution over the codewords and models the information source behavior. As in the previous Sections, we shall assume the source distribution μ_0 to be uniform over the codewords, cf. Eq. (6.4). On the other hand, the codewords themselves have been constructed using the single-bit distribution $P(x)$.

We shall first analyze the RCE for a generic distribution $P(x)$, under word MAP decoding. The main result is:

Theorem 6.1 *Consider communication over a binary input discrete memoryless channel with transition probability $Q(y|x)$, using a code from the RCE with input bit distribution $P(x)$ and word MAP decoding. If the code rate is smaller than the mutual information $I_{X,Y}$ between two random variables X, Y with joint distribution $P(x)Q(y|x)$, then the block error rate vanishes in the large blocklength limit.*

Using this result, one can optimize the ensemble performances over the choice of the distribution $P(\cdot)$. More precisely, we maximixe the achievable rate for error-free

communication: $I_{X,Y}$. The corresponding optimal distribution $P^*(\cdot)$ depends upon the channel: it is the best distribution adapted to the channel. Since the channel capacity is in fact defined as the maximum mutual information between channel input and channel output, cf. Eq. (1.38), the RCE with input bit distribution $P^*(\cdot)$ allows to communicate error-free up to channel capacity. The above Theorem implies therefore the 'direct part' of Shannon's theorem 1.23.

Proof: Assume that the codeword $\underline{x}^{(0)}$ is transmitted through the channel and the message $\underline{y} \in \mathcal{Y}^N$ is received. The decoder constructs the probability for $\underline{x}$ to be the channel input, conditional to the output $\underline{y}$, see Eq. (6.5). Word MAP decoding consists in minimizing the cost function

$$E(\underline{x}) = -\sum_{i=1}^{N} \log_2 Q(y_i|x_i) \tag{6.29}$$

over the codewords $\underline{x} \in \mathfrak{C}_N$ (note that we use here natural logarithms). Decoding will be successful if and only if the minimum of $E(\underline{x})$ is realized over the transmitted codeword $\underline{x}^{(0)}$. The problem consists therefore in understanding the behavior of the 2^M random variables $E(\underline{x}^{(0)}), \ldots, E(\underline{x}^{(2^M-1)})$.

Once more, it is necessary to single out $E(\underline{x}^{(0)})$. This is the sum of N i.i.d. random variables $-\log Q(y_i|x_i^{(0)})$, and it is therefore well approximated by its mean

$$\mathbb{E}\, E(\underline{x}^{(0)}) = -N \sum_{x,y} P(x)Q(y|x) \log_2 Q(y|x) = N H_{Y|X}\,. \tag{6.30}$$

In particular $(1-\delta)NH_{Y|X} < E(\underline{x}^{(0)}) < (1+\delta)NH_{Y|X}$ with probability approaching one as $N \to \infty$.

As for the 2^M-1 incorrect codewords, the corresponding log-likelihoods $E(\underline{x}^{(1)}), \ldots, E(\underline{x}^{(2^M-1)})$ are i.i.d. random variables. We can therefore estimate the smallest among them by following the approach developed for the REM and already applied to the RCE on the BSC. In Appendix 6.7, we prove the following large deviation result on the distribution of these variables:

Lemma 6.2 *Let $\varepsilon_i = E(\underline{x}^{(i)})/N$. Then $\varepsilon_1, \ldots, \varepsilon_{2^M-1}$ are i.i.d. random variables and their distribution satisfy a large deviation principle of the form $\mathbb{P}(\varepsilon) \doteq 2^{-N\psi(\varepsilon)}$. The rate function is given by:*

$$\psi(\varepsilon) \equiv \min_{\{p_y(\cdot)\}\in\mathfrak{P}_\varepsilon} \left[\sum_y Q(y) D(p_y||P) \right]\,, \tag{6.31}$$

where the minimum is taken over the set of probability distributions $\{p_y(\cdot),\ y \in \mathcal{Y}\}$ in the subspace $\mathfrak{P}_\varepsilon$ defined by the constraint:

$$\varepsilon = -\sum_{xy} Q(y) p_y(x) \log_2 Q(y|x)\,, \tag{6.32}$$

and we defined $Q(y) \equiv \sum_x Q(y|x)P(x)$.

The solution of the minimization problem formulated in this lemma is obtained through a standard Lagrange multiplier technique:

$$p_y(x) = \frac{1}{z(y)}\, P(x)Q(y|x)^\gamma\,, \tag{6.33}$$

where the (ε dependent) constants $z(y)$ and γ are chosen in order to verify the normalizations $\sum_x p_y(x) = 1$ for all $y \in \mathcal{Y}$, and the constraint (6.32).

The rate function $\psi(\varepsilon)$ is convex with a global minimum (corresponding to $\gamma = 0$) at $\varepsilon_* = -\sum_{x,y} P(x)Q(y)\log_2 Q(y|x)$ where its value is $\psi(\varepsilon_*) = 0$. This implies that, with high probability, all incorrect codewords will have costs $E(\underline{x}^{(i)}) = N\varepsilon$ in the range $\varepsilon_{\min} - \delta \le \varepsilon \le \varepsilon_{\max} + \delta$ for all $\delta > 0$, $\varepsilon_{\min}$ and $\varepsilon_{\max}$ being the two solutions of $\psi(\varepsilon) = R$. Moreover, for any ε inside the interval, the number of codewords with $E(\underline{x}^{(i)}) \simeq N\varepsilon$ is exponentially large, and close to $2^{NR - N\psi(\varepsilon)}$. So with high probability the incorrect codeword with minimum cost has a cost close to $N\varepsilon_{\min}$, while the correct codeword has cost close to $NH_{Y|X}$. Therefore MAP decoding will find the correct codeword if and only if $H_{Y|X} < \varepsilon_{\min}$.

Let us now show that the condition $H_{Y|X} < \varepsilon_{\min}$ is in fact equivalent to $R < I_{X,Y}$. It turns out that the value $\varepsilon = H_{Y|X}$ is obtained using $\gamma = 1$ in Eq. (6.33) and therefore $p_y(x) = P(x)Q(y|x)/Q(y)$. The corresponding value of the rate function is $\psi(\varepsilon = H_{Y|X}) = H_Y - H_{Y|X} = I_{Y|X}$. The condition for error free communication, $H_{Y|X} < \varepsilon_{\min}$, can thus be rewritten as $R < \psi(H_{Y|X})$, or $R < I_{X,Y}$. □

Example 6.3 Reconsider the BSC with flip probability p. We have

$$E(\underline{x}) = -(N - d(\underline{x},\underline{y}))\,\log(1-p) - d(\underline{x},\underline{y})\,\log p\,. \tag{6.34}$$

Up to a rescaling the cost coincides with the Hamming distance from the received message. If we take $P(\mathtt{0}) = P(\mathtt{1}) = 1/2$, the optimal types are, cf. Eq. (6.33),

$$p_0(\mathtt{1}) = 1 - p_0(\mathtt{0}) = \frac{p^\gamma}{(1-p)^\gamma + p^\gamma}\,, \tag{6.35}$$

and analogously for $p_1(x)$. The corresponding cost is

$$\varepsilon = -(1-\delta)\log(1-p) - \delta\log p\,, \tag{6.36}$$

where we defined $\delta = p^\gamma/[(1-p)^\gamma + p^\gamma]$. The large deviations rate function is given, parametrically, by $\psi(\varepsilon) = \log 2 - \mathcal{H}(\delta)$. The reader will easily recognize the results already obtained in the previous section.

Exercise 6.7 Consider communication over a discrete memoryless channel with finite input output alphabets $\mathcal{X}$, and $\mathcal{Y}$, and transition probability $Q(y|x)$, $x \in \mathcal{X}$, $y \in \mathcal{Y}$. Check that the above proof remains valid in this context.

6.5 Geometry again: sphere packing

Coding has a lot to do with the optimal packing of spheres, which is a mathematical problem of considerable interest in various branches of science. Consider for instance the communication over a BSC with flip probability p. A code of rate R and blocklength N consists of 2^{NR} points $\{\underline{x}^{(1)} \cdots \underline{x}^{(2^{NR})}\}$ in the hypercube $\{0,1\}^N$. To each possible channel output $\underline{y} \in \{0,1\}^N$, the decoder associates one of the codewords $\underline{x}^{(i)}$. Therefore we can think of the decoder as realizing a partition of the Hamming space in 2^{NR} decision regions $\mathfrak{D}^{(i)}$, $i \in \{1 \dots 2^{NR}\}$, each one associated to a distinct codeword. If we require each decision region $\{\mathfrak{D}^{(i)}\}$ to contain a sphere of radius ρ, the resulting code is *guaranteed* to correct *any* error pattern such that less than ρ bits are flipped. One often defines the **minimum distance** of a code as the smallest distance between any two codewords If a code has minimal distance d, the Hamming spheres of radius $\rho = \lfloor (d-1)/2 \rfloor$ don't overlap and the code can correct ρ errors, whatever their positions.

6.5.1 The densest packing of Hamming spheres

We are thus led to consider the general problem of sphere packing on the hypercube $\{0,1\}^N$. A (Hamming) sphere of center $\underline{x}^{(0)}$ and radius r is defined as the set of points $\underline{x} \in \{0,1\}^N$, such that $d(\underline{x}, \underline{x}^{(0)}) \leq r$. A packing of spheres of radius r and cardinality $\mathcal{N}_{\mathcal{S}}$ is specified by a set of centers $\underline{x}_1, \dots, \underline{x}_{\mathcal{N}_{\mathcal{S}}} \in \{0,1\}^N$, such that the spheres of radius r centered in these points are disjoint. Let $\mathcal{N}_N^{\max}(\delta)$ be the maximum cardinality of a packing of spheres of radius $N\delta$ in $\{0,1\}^N$. We define the corresponding rate as $R_N^{\max}(\delta) \equiv N^{-1} \log_2 \mathcal{N}_N^{\max}(\delta)$ and would like to compute this quantity in the infinite-dimensional limit

$$R^{\max}(\delta) \equiv \lim \sup_{N \to \infty} R_N^{\max}(\delta) \,. \tag{6.37}$$

The problem of determining the function $R^{\max}(\delta)$ is open: only upper and lower bounds are known. Here we shall derive the simplest of these bounds:

Proposition 6.4

$$1 - \mathcal{H}_2(2\delta) \leq R^{\max}(\delta) \leq 1 - \mathcal{H}_2(\delta) \tag{6.38}$$

The lower bound is often called the Gilbert-Varshamov *bound, the upper bound is called the* Hamming *bound.*

Proof: Lower bounds can be proved by analyzing good packing strategies. A simple such strategy consists in taking the sphere centers as 2^{NR} random points with uniform probability in the Hamming space. The minimum distance between any couple of points must be larger than $2N\delta$. It can be estimated by defining the distance enumerator $\mathcal{M}_2(d)$ which counts how many couples of points have distance d. It is straightforward to show that, if $d = 2N\delta$ and δ is kept fixed as $N \to \infty$:

$$\mathbb{E}\, \mathcal{M}_2(d) = \binom{2^{NR}}{2} 2^{-N} \binom{N}{d} \doteq 2^{N[2R-1+\mathcal{H}_2(2\delta)]} \,. \tag{6.39}$$

As long as $R < [1 - \mathcal{H}_2(2\delta)]/2$, the exponent in the above expression is negative. Therefore, by Markov inequality, the probability of having any couple of centers ar a distance smaller than 2δ is exponentially small in the size. This implies that

$$R^{\max}(\delta) \geq \frac{1}{2}[1 - \mathcal{H}_2(2\delta)]\,. \tag{6.40}$$

A better lower bound can be obtained by a closer examination of the above (random) packing strategy. In Sec. 6.2 we derived the following result. If 2^{NR} points are chosen from the uniform distribution in the Hamming space $\{0,1\}^N$, and one of them is considered, with high probability its closest neighbour is at a Hamming distance close to $N\delta_{\rm GV}(R)$. In other words, if we draw around each point a sphere of radius δ, with $\delta < \delta_{\rm GV}(R)/2$, and one of the spheres is selected randomly, with high probability it will not intersect any other sphere. This remark suggests the following trick (sometimes called **expurgation** in coding theory). Go through all the spheres one by one and check if it intersects any other one. If the answer is positive, simply eliminate the sphere. This reduces the cardinality of the packing, but only by a fraction approaching 0 as $N \to \infty$: the packing rate is thus unchanged. As $\delta_{\rm GV}(R)$ is defined by $R = 1 - \mathcal{H}_2(\delta_{\rm GV}(R))$, this proves the lower bound in (6.38).

The upper bound can be obtained from the fact that the total volume occupied by the spheres is not larger than the volume of the hypercube. If we denote by $\Lambda_N(\delta)$ the volume of an N-dimensional Hamming sphere of radius $N\delta$, we get $\mathcal{N}_{\mathcal{S}}\,\Lambda_N(\delta) \leq 2^N$. Since $\Lambda_N(\delta) \doteq 2^{N\mathcal{H}_2(\delta)}$, this implies the upper bound in (6.38). □

Better upper bounds can be derived using more sophisticated mathematical tools. An important result of this type is the so-called *linear programming bound*:

$$R^{\max}(\delta) \leq \mathcal{H}_2(1/2 - \sqrt{2\delta(1-2\delta)})\,, \tag{6.41}$$

whose proof goes beyond our scope. On the other hand, no better lower bound than the Gilbert-Varshamov result is known. It is a widespread conjecture that this bound is indeed tight: in high dimension there is no better way to pack spheres than placing them randomly and expurgating the small fraction of them that are 'squeezed'. The various bounds are shown in Fig. 6.6.

Exercise 6.8 Derive two simple alternative proofs of the Gilbert-Varshamov bound using the following hints:

(*a*) Given a constant $\overline{\delta}$, let's look at all the 'dangerous' couples of points whose distance is smaller than $2N\overline{\delta}$. For each dangerous couple, we can expurgate one of its two points. The number of points expurgated is smaller or equal than the number of dangerous couples, which can be bounded using $\mathbb{E}\,\mathcal{M}_2(d)$. What is the largest value of $\overline{\delta}$ such that this expurgation procedure does not reduce the rate?

(*b*) Construct a packing $\underline{x}_1 \dots \underline{x}_{\mathcal{N}}$ as follows. The first center $\underline{x}_1$ can be placed anywhere in $\{0,1\}^N$. The second one is everywhere outside a sphere of radius $2N\delta$ centered in $\underline{x}^{(0)}$. In general the i-th center $\underline{x}_i$ can be at any point outside the spheres centered in $\underline{x}_1 \dots \underline{x}_{i-1}$. This procedures stops when the spheres of radius $2N\delta$ cover all the space $\{0,1\}^N$, giving a packing of cardinality $\mathcal{N}$ equal to the number of steps and radius $N\delta$.

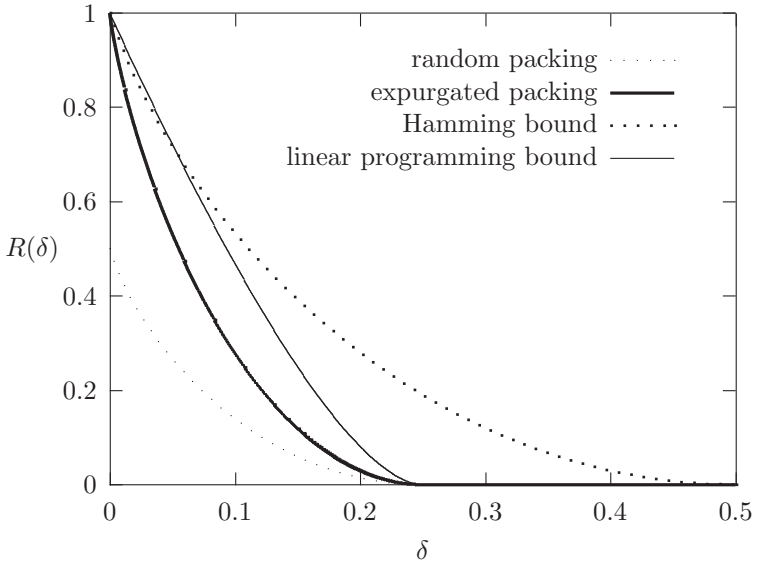

Fig. 6.6 Upper and lower bounds on the maximum packing rate $R^{\max}(\delta)$ of Hamming spheres of radius $N\delta$. Random packing and expurgated random packing provide lower bounds. The Hamming and linear programming bounds are upper bounds.

6.5.2 Sphere packing and decoding over the BSC

Let us now see the consequences of Proposition 6.4 for coding over the BSC. If the transmitted codeword is $\underline{x}^{(i)}$, the channel output will be (with high probability) at a distance from $\underline{x}^{(i)}$ close to Np. Clearly $R \leq R^{\max}(p)$ is a necessary and sufficient condition for the existence of a code which corrects *any* error pattern such that less than Np bits are flipped. Notice that this correction criterion is much stronger than requiring a vanishing (bit or block) error rate. The direct part of Shannon theorem shows the existence of codes with a vanishing (as $N \to \infty$) block error probability for $R < 1 - \mathcal{H}_2(p) = C_{\rm BSC}(p)$. As shown by the linear programming bound in Fig. 6.6 $C_{\rm BSC}(p)$ lies above $R^{\max}(p)$ for large enough p. Therefore, for such values of p, there is a non-vanishing interval of rates $R^{\max}(p) < R < C_{\rm BSC}(p)$ such that one can correct Np errors with high probability but one cannot correct *all* error patterns involving that many bits.

Let us show, for the BSC case, that the condition $R < 1 - \mathcal{H}_2(p)$ is actually a necessary one for achieving zero block error probability (this is nothing but the converse part of Shannon channel coding theorem 1.23).

Define $\mathrm{P_B}(k)$ the block error probability under the condition that k bits are flipped by the channel. If the codeword $\underline{x}^{(i)}$ is transmitted, the channel output lies on the border of a Hamming sphere of radius k centered in $\underline{x}^{(i)}$: $\partial B_i(k) \equiv \{\underline{z} \,:\, d(\underline{z}, \underline{x}^{(i)}) = k\}$. Therefore

$$\mathrm{P_B}(k) = \frac{1}{2^{NR}} \sum_{i=1}^{2^{NR}} \left[1 - \frac{|\partial B_i(k) \cap \mathfrak{D}^{(i)}|}{|\partial B_i(k)|} \right] \geq \tag{6.42}$$

$$\geq 1 - \frac{1}{2^{NR}} \sum_{i=1}^{2^{NR}} \frac{|\mathfrak{D}^{(i)}|}{|\partial B_i(k)|} \,. \tag{6.43}$$

For a typical channel realization k is close to Np, and $|\partial B_i(Np)| \doteq 2^{N\mathcal{H}_2(p)}$. Since $\{\mathfrak{D}^{(i)}\}$ is a partition of $\{0,1\}^N$, $\sum_i |\mathfrak{D}^{(i)}| = 2^N$. We deduce that, for any $\varepsilon > 0$, and large enough N:

$$\mathrm{P_B} \geq 1 - 2^{N(1-R-\mathcal{H}_2(p)+\varepsilon)} \,, \tag{6.44}$$

and thus reliable communication is possible only if $R \leq 1 - \mathcal{H}_2(p)$.

6.6 Other random codes

A major drawback of the random code ensemble is that specifying a particular code (an element of the ensemble) requires $N2^{NR}$ bits. This information has to be stored somewhere when the code is used in practice and the memory requirement is soon beyond the hardware capabilities. A much more compact specification is possible for the **random linear code (RLC)** ensemble. In this case the encoder is required to be a linear map, and any such map is equiprobable. Concretely, the code is fully specified by a $N \times M$ binary matrix $\mathbb{G} = \{G_{ij}\}$ (the **generator matrix**) and encoding is left multiplication by $\mathbb{G}$:

$$\underline{x} : \{0,1\}^M \to \{0,1\}^N \,, \tag{6.45}$$

$$\underline{z} \mapsto \mathbb{G}\,\underline{z} \,, \tag{6.46}$$

where the multiplication has to be carried modulo 2. Endowing the set of linear codes with uniform probability distribution is essentially equivalent to assuming the entries of $\mathbb{G}$ to be i.i.d. random variables, with $G_{ij} = 0$ or 1 with probability $1/2$. Notice that only MN bits are required for specifying a code within this ensemble.

Exercise 6.9 Consider a linear code with $N = 4$ and $|\mathfrak{C}| = 8$ defined by

$$\mathfrak{C} = \{(z_1 \oplus z_2,\, z_2 \oplus z_3,\, z_1 \oplus z_3,\, z_1 \oplus z_2 \oplus z_3) \mid \; z_1, z_2, z_3 \in \{0,1\}\} \,, \tag{6.47}$$

where we denoted by $\oplus$ the sum modulo 2. For instance $(0110) \in \mathfrak{C}$ because we can take $z_1 = 1$, $z_2 = 1$ and $z_3 = 0$, but $(0010) \notin \mathfrak{C}$. Compute the distance enumerator for $\underline{x}^{(0)} = (0110)$.

It turns out that the RLC has extremely good performances. As the original Shannon ensemble, it allows to communicate error-free below capacity. Moreover, the rate at which the block error probability $\mathrm{P_B}$ vanishes is faster for the RLC than for the RCE. This justifies the considerable effort devoted so far to the design and analysis of specific ensembles of linear codes satisfying additional computational requirements. We shall discuss some among the best such codes in the following Chapters.

6.7 A remark on coding theory and disordered systems

We would like to stress here the fundamental similarity between the analysis of random code ensembles and the statistical physics of disordered systems. As should be already clear, there are several sources of randomness in coding:

- First of all, the code used is chosen randomly from an ensemble. This was the original idea used by Shannon to prove the channel coding theorem.
- The codeword to be transmitted is chosen with uniform probability from the code. This hypothesis is supported by the source-channel separation theorem.
- The channel output is distributed, once the transmitted codeword is fixed, according to a probabilistic process which accounts for the channel noise.
- Once all the above elements are given, one is left with the decoding problem. As we have seen in Sec. 6.3.3, both classical MAP decoding strategies and finite-temperature decoding can be defined in a unified frame. The decoder constructs a probability distribution $\mu_{y,\beta}(\underline{x})$ over the possible channel inputs, and estimates its single bit marginals $\mu^{(i)}_{y,\beta}(x_i)$. The decision on the i-th bit depends upon the distribution $\mu^{(i)}_{y,\beta}(x_i)$.

The analysis of a particular coding system can therefore be regarded as the analysis of the properties of the distribution $\mu_{y,\beta}(\underline{x})$ when the code, the transmitted codeword and the noise realization are distributed as explained above.

In other words, we are distinguishing two levels of randomness: on the first level we deal with the first three sources of randomness, and on the second level we use the distribution $\mu_{y,\beta}(\underline{x})$. The deep analogy with the theory of disordered system should be clear at this point. The code, channel input, and noise realization play the role of *quenched disorder* (the sample), while the distribution $\mu_{y,\beta}(\underline{x})$ is the analog of the *Boltzmann's distribution*. In both cases the problem consists in studying the properties of a probability distribution which is itself a random object.

Appendix: Proof of Lemma 6.2

We estimate (to the leading exponential order in the large N limit) the probability $\mathbb{P}_N(\varepsilon)$ for one of the incorrect codewords, $\underline{x}$, to have cost $E(\underline{x}) = N\varepsilon$. The channel output $\underline{y} = (y_1 \cdots y_N)$ is a sequence of N i.i.d. symbols distributed according to

$$Q(y) \equiv \sum_x Q(y|x)P(x)\,, \tag{6.48}$$

and the cost can be rewritten as:

$$\begin{aligned} E(\underline{x}) &\equiv -\sum_{i=1}^{N} \log Q(y_i|x_i) \\ &= -N \sum_{x,y} Q(y) \log Q(y|x) \frac{1}{NQ(y)} \sum_{i=1}^{N} \mathbb{I}(x_i = x, y_i = y)\,. \end{aligned} \tag{6.49}$$

There are approximatively $NQ(y)$ positions i such that $y_i = y$, for $y \in \mathcal{Y}$. We assume that there are *exactly* $NQ(y)$ such positions, and that $NQ(y)$ is an integer (of course

this hypothesis is in general false: it is a routine exercise, left to the reader, to show that it can be avoided with a small technical detour). Furthermore we introduce

$$p_y(x) \equiv \frac{1}{NQ(y)} \sum_{i=1}^{N} \mathbb{I}(x_i = x,\ y_i = y)\,. \tag{6.50}$$

Under the above assumptions the function $p_y(x)$ is a probability distribution over $x \in \{0,1\}$ for each $y \in \mathcal{Y}$. Looking at the subsequence of positions i such that $y_i = y$, it counts the fraction of the x_i's such that $x_i = x$. In other words $p_y(\cdot)$ is the type of the subsequence $\{x_i | y_i = y\}$. Because of Eq. (6.49), the cost is written in terms of these types as follows

$$E(\underline{x}) = -N \sum_{xy} Q(y) p_y(x) \log Q(y|x)\,. \tag{6.51}$$

Therefore $E(\underline{x})$ depends upon $\underline{x}$ uniquely through the types $\{p_y(\cdot) : y \in \mathcal{Y}\}$, and this dependence is linear in $p_y(x)$. Moreover, according to our definition of the RCE, $x_1, \dots, x_N$ are i.i.d. random variables with distribution $P(x)$. The probability $\mathbb{P}_N(\varepsilon)$ that $E(\underline{x})/N = \varepsilon$ can therefore be deduced from the Corollary **??**. To the leading exponential order, we get

$$\mathbb{P}_N(\varepsilon) \doteq \exp\{-N\psi(\varepsilon)\log 2\}\,, \tag{6.52}$$

$$\psi(\varepsilon) \equiv \min_{p_y(\cdot)} \left[\sum_y Q(y) D(p_y||P) \ \text{ s.t. } \ \varepsilon = -\sum_{xy} Q(y) p_y(x) \log_2 Q(y|x) \right]\,. \tag{6.53}$$

Notes

The random code ensemble dates back to Shannon (Shannon, 1948) who used it (somehow implicitely) in his proof of the channel coding thorem. A more explicit (and complete) proof was provided by Gallager in (Gallager, 1965). The reader can find alternative proofs in standard textbooks such as (Cover and Thomas, 1991; Csiszár and Körner, 1981; Gallager, 1968).

The distance enumerator is a feature extensively investigated in coding theory. We refer for instance to (Csiszár and Körner, 1981; Gallager, 1968). A treatment of the random code ensemble in analogy with the random energy model was presented in (Montanari, 2001). More detailed results in the same spirit can be found in (Barg and Forney, 2002; Forney and Montanari, 2001). The analogy between coding theory and the statistical physics of disordered systems was put forward by Sourlas (Sourlas, 1989). Finite temperature decoding has been introduced in (Rujan, 1993).

A key ingredient of our analysis was the assumption, already mentioned in Sec. 1.6.2, that any codeword is *a priori* equiprobable. The fundamental motivation for such an assumption is the source-channel separation theorem. In simple terms: one does not loose anything in constructing an encoding system in two blocks. First an ideal source code compresses the data produced by the information source and outputs a sequence of i.i.d. unbiased bits. Then a channel code adds redundancy to this sequence in order to contrast the noise on the channel. The theory of error correcting codes focuses on

the design and analysis of this second block, leaving the first one to source coding. The interested reader may find a proofs of the separation theorem in (Cover and Thomas, 1991; Csiszár and Körner, 1981; Gallager, 1968).

Sphere packing is a classical problem in mathematics, with applications in various branches of science. The book (Conway and Sloane, 1998) provides both a very good introduction and some far reaching results on this problem and its connections, in particular to coding theory. Finding the densest packing of spheres in $\mathbb{R}^n$ is an open problem when $n \geq 4$.

9
Factor graphs and graph ensembles

Systems involving a large number of simple variables with mutual dependencies (or constraints, or interactions) appear recurrently in several fields of science. It is often the case that such dependencies can be 'factorized' in a non-trivial way, and distinct variables interact only 'locally'. In statistical physics, the fundamental origin of such a property can be traced back to the locality of physical interactions. In computer vision it is due to the two dimensional character of the retina and the locality of reconstruction rules. In coding theory it is a useful property for designing a system with fast encoding/decoding algorithms. This important structural property plays a crucial role in many interesting problems.

There exist several possibilities for expressing graphically the structure of dependencies among random variables: graphical models, Bayesian networks, dependency graphs, normal realizations, etc. We adopt here the *factor graph* language, because of its simplicity and flexibility.

As argumented in the previous Chapters, we are particularly interested in *ensembles* of probability distributions. These may emerge either from ensembles of error correcting codes, or in the study of disordered materials, or, finally, when studying random combinatorial optimization problems. Problems drawn from these ensembles are represented by factor graphs which are themselves *random*. The most common examples are random hyper-graphs, which are a simple generalization of the well known random graphs.

Section 9.1 introduces factor graphs and provides a few examples of their utility. In Sec. 9.2 we define some standard ensembles of random graphs and hyper-graphs. We summarize some of their important properties in Sec. 9.3. One of the most surprising phenomena in random graph ensembles is the sudden appearance of a 'giant' connected component as the number of edges crosses a threshold. This is the subject of Sec. 9.4. Finally, in Sec. 9.5 we describe the local structure of large random factor graphs.

9.1 Factor graphs

9.1.1 Definitions and general properties

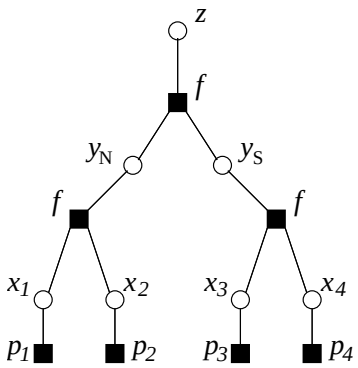

Fig. 9.1 Factor graph representation of the electoral process described in Example 1.

Example 9.1 We begin with a toy example. A country elects its president among two candidates $\{A, B\}$ according to the following peculiar system. The country is divided into four regions $\{1, 2, 3, 4\}$, grouped in two states: North (regions 1 and 2), and South (3 and 4). Each of the regions chooses its favorite candidate according to popular vote: we call her $x_i \in \{A, B\}$, with $i \in \{1, 2, 3, 4\}$. Then a North candidate $y_{\rm N}$, and a South candidate $y_{\rm S}$ are decided according to the following rule. If the preferences x_1 and x_2 in regions 1 and 2 agree, then $y_{\rm N}$ takes this same value. If they don't agree $y_{\rm N}$ is decided according to a fair coin trial. The same procedure is adopted for the choice of $y_{\rm S}$, given x_3, x_4. Finally, the president $z \in \{A, B\}$ is decided on the basis of the choices $y_{\rm N}$ and $y_{\rm S}$ in the two states using the same rule as inside each state.

A polling institute has obtained fairly good estimates of the probabilities $p_i(x_i)$ for the popular vote in each region i to favor the candidate x_i. They ask you to calculate the odds for each of the candidates to become the president.

It is clear that the electoral procedure described above has important 'factorization' properties. More precisely, the probability distribution for a given realization of the random variables $\{x_i\}$, $\{y_j\}$, z has the form:

$$P(\{x_i\}, \{y_j\}, z) = f(z, y_{\rm N}, y_{\rm S})\, f(y_{\rm N}, x_1, x_2)\, f(y_{\rm S}, x_3, x_4) \prod_{i=1}^{4} p_i(x_i)\,. \qquad (9.1)$$

We leave it to the reader to write explicit forms for the function f. The election process, as well as the above probability distribution, can be represented graphically as in Fig. 9.1. Can this particular structure be exploited when computing the chances for each candidate to become president?

Abstracting from the previous example, let us consider a set of N variables $x_1, \ldots, x_N$ taking values in a finite alphabet $\mathcal{X}$. We assume that their joint probability distribution takes the form

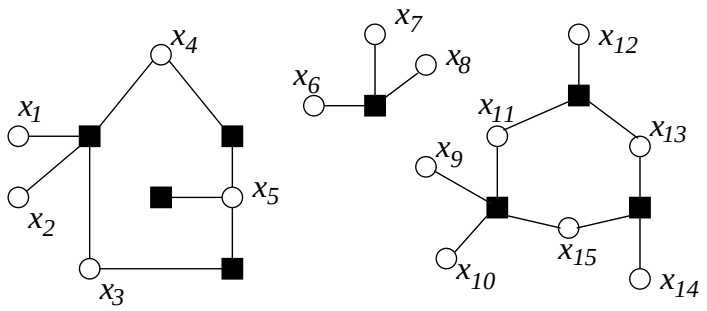

Fig. 9.2 A generic factor graph is formed by several connected components. Variables belonging to distinct components (for instance x_3 and x_{15} in the graph above) are statistically independent.

$$P(\underline{x}) = \frac{1}{Z}\prod_{a=1}^{M}\psi_a(\underline{x}_{\partial a})\,. \tag{9.2}$$

Here we use the shorthands $\underline{x} \equiv \{x_1,\dots,x_N\}$, and $\underline{x}_{\partial a} \equiv \{x_i \,|\, i \in \partial a\}$, where $\partial a \subseteq [N]$. The set of indices ∂a, with $a \in [M]$, has size $k_a \equiv |\partial a|$. When necessary, we shall use the notation $\{i^a_1,\dots,i^a_{k_a}\} \equiv \partial a$ to denote the variable indices which correspond to the factor a, and $\underline{x}_{i^a_1,\dots,i^a_{k_a}} \equiv \underline{x}_{\partial a}$ for the corresponding variables. The **compatibility functions** $\psi_a : \mathcal{X}^{k_a} \to \mathbb{R}$ are non-negative, and Z is a positive constant. In order to completely determine the form (9.2), we should precise both the functions $\psi_a(\,\cdot\,)$, and an ordering among the indices in ∂a. In practice this last specification will be always clear from the context.

Factor graphs provide a graphical representations of distributions of the form (9.2) which are also denominated undirected **graphical models**. The factor graph for the distribution (9.2) contains two types of nodes: N **variable nodes**, each one associated with a variable x_i (represented by circles); M **function nodes**, each one associated with a function ψ_a (squares). An edge joins the variable node i and the function node a if the variable x_i is among the arguments of $\psi_a(\underline{x}_{\partial a})$ (in other words if $i \in \partial a$). The set of function nodes that are adjacent to (share an edge with) the variable node i, is denoted as ∂i. The graph is bipartite: an edge always joins a variable node to a function nodes. The reader will easily check that the graph in Fig. 9.1 is indeed the factor graph corresponding to the factorized form (9.1). The degree of a variable node $|\partial i|$, or of a factor node $|\partial a|$, is defined as usual as the number of edges incident on it. In order to avoid trivial cases, we will assume $|\partial a| \geq 1$ for any factor node a. The basic property of the probability distribution (9.2), encoded in its factor graph, is that two 'well separated' variables interact uniquely through those variables which are interposed between them. A precise formulation of this intuition is given by the following observation, named the **global Markov property**:

Proposition 9.2 *Let $A, B, S \subseteq [N]$ be three disjoint subsets of the variable nodes, and denote by $\underline{x}_A$, $\underline{x}_B$ and $\underline{x}_S$ the corresponding sets of variables. If S 'separates' A and B (i.e., if there is no path in the factor graph joining a node of A to a node of B*

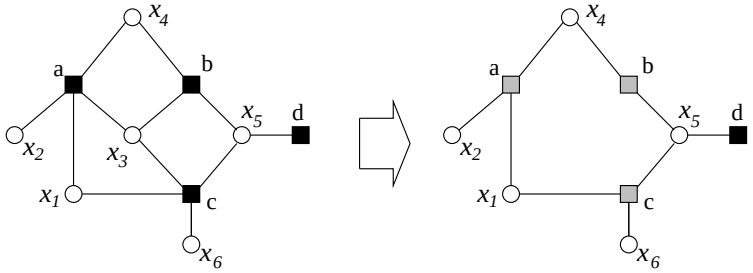

Fig. 9.3 The action of conditioning on the factor graph. The probability distribution on the left has the form $P(\underline{x}_{1\dots6}) \propto f_a(\underline{x}_{1\dots4}) f_b(\underline{x}_{3,4,5}) f_c(\underline{x}_{1,3,5,6}) f_d(\underline{x}_5)$. After conditioning on x_3, we get $P(\underline{x}_{1\dots6}|x_3 = x_*) \propto f'_a(\underline{x}_{1,2,4}) f'_b(\underline{x}_{4,5}) f'_c(\underline{x}_{1,5,6}) f_d(\underline{x}_5)$. Notice that the functions $f'_a(\cdot)$, $f'_b(\cdot)$, $f'_c(\cdot)$ (gray nodes on the right) are distinct from $f_a(\cdot)$, $f_b(\cdot)$, $f_c(\cdot)$ and depend upon the value x_*.

without passing through S) then

$$P(\underline{x}_A, \underline{x}_B|\underline{x}_S) = P(\underline{x}_A|\underline{x}_S)\, P(\underline{x}_B|\underline{x}_S)\,. \tag{9.3}$$

In such a case the variables $\underline{x}_A, \underline{x}_B$ are said to be conditionally independent.

Proof: It is easy to provide a 'graphical' proof of this statement. Notice that, if the factor graph is disconnected, then variables belonging to distinct components are independent, cf. Fig. 9.2. Conditioning upon a variable x_i is equivalent to eliminating the corresponding variable node from the graph and modifying the adjacent function nodes accordingly, cf. Fig. 9.3. Finally, when conditioning upon $\underline{x}_S$ as in Eq. (9.3), the factor graph gets split in such a way that A and B belong to distinct components. We leave to the reader the exercise of filling the details. □

It is natural to wonder whether any probability distribution which is 'globally Markov' with respect to a given graph can be written in the form (9.2). In general, the answer is negative, as can be shown on a simple example. Consider the small factor graph in Fig. (9.4). The global Markov property has a non trivial content only for the following choice of subsets: $A = \{1\}$, $B = \{2, 3\}$, $S = \{4\}$. The most general probability distribution such that x_1 is independent from $\{x_2, x_3\}$ conditionally to x_4 is of the type $f_a(x_1, x_4) f_b(x_2, x_3, x_4)$. The probability distribution encoded by the factor graph is a special case where $f_b(x_2, x_3, x_4) = f_c(x_2, x_3) f_d(x_3, x_4) f_e(x_4, x_2)$.

The factor graph of our counterexample, Fig. 9.4, has a peculiar property: it contains a subgraph (the one with variables $\{x_2, x_3, x_4\}$) such that, for any pair of variable nodes, there is a function node adjacent to both of them. We call any factor subgraph possessing this property a **clique** (its definition generalizes the notion of clique in usual graphs). It turns out that, once one gets rid of cliques, the converse of Proposition 9.2 can be proved. We shall 'get rid' of cliques by completing the factor graph. Given a factor graph F, its **completion** $\overline{F}$ is obtained by adding one factor node for each clique in the graph and connecting it to each variable node in the clique and to no other node.

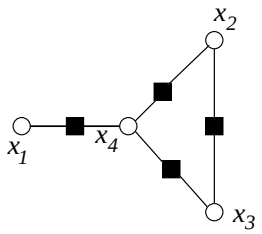

Fig. 9.4 A factor graph with four variables. $\{x_1\}$ and $\{x_2, x_3\}$ are independent conditionally to x_4. The set of variables $\{x_2, x_3, x_4\}$ and the three function nodes connecting two points in this set form a clique.

Theorem 9.3. (Hammersley-Clifford) *Let $P(\cdot)$ be a strictly positive probability distributions over the variables $\underline{x} = (x_1, \dots, x_N) \in \mathcal{X}^N$, satisfying the global Markov property (9.3) with respect to a factor graph F. Then P can be written in the factorized form (9.2), with respect to the completed graph $\overline{F}$.*

Roughly speaking: the only assumption behind the factorized form (9.2) is the rather weak notion of locality encoded by the global Markov property. This may serve as a general justification for studying probability distributions having a factorized form. Notice that the positivity hypothesis $P(x_1, \dots, x_N) > 0$ is not just a technical assumption: there exist counterexamples to the Hammersley-Clifford theorem if P is allowed to vanish.

9.1.2 Examples

Let us look at a few examples.

We start with the Markov chains. The random variables $X_1, \dots, X_N$ taking values in the finite state space $\mathcal{X}$ form a **Markov chain of order** r (with $r < N$) if

$$P(x_1 \dots x_N) = P_0(x_1 \dots x_r) \prod_{t=r}^{N-1} w(x_{t-r+1} \dots x_t \to x_{t+1}) \,, \tag{9.4}$$

for some non-negative transition probabilities $\{w(x_{-r} \dots x_{-1} \to x_0)\}$, and initial condition $P_0(x_1 \dots x_r)$, satisfying the normalization conditions

$$\sum_{x_1 \dots x_r} P_0(x_1 \dots x_r) = 1 \,, \qquad \sum_{x_0} w(x_{-r} \dots x_{-1} \to x_0) = 1 \,. \tag{9.5}$$

The parameter r is the 'memory range' of the chain. Ordinary Markov chains have $r = 1$. Higher order Markov chains allow to model more complex phenomena. For instance, in order to get a reasonable probabilistic model of the English language with the usual alphabet $\mathcal{X} = \{$a,b,... z, blank$\}$ as state space, it is reasonable to choose r of the order of the average word length.

It is clear that Eq. (9.4) is a particular case of the factorized form (9.2). The corresponding factor graph includes N variable nodes, one for each variable x_i, $N - r$ function nodes, one for each of the factors $w(\cdot)$, and one function node for the initial condition $P_0(\cdot)$. In Fig. 9.5 we present a small example with $N = 6$ and $r = 2$.

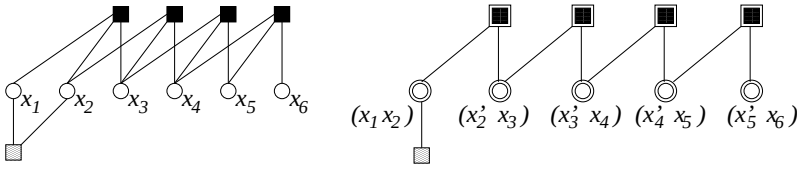

Fig. 9.5 On the left: factor graph for a Markov chain of length $N = 6$ and memory range $r = 2$. On the right: by adding auxiliary variables, the same probability distribution can be written as a Markov chain with memory range $r = 1$.

Exercise 9.1 Show that a Markov chain with memory r and state space $\mathcal{X}$ can always be rewritten as a Markov chain with memory 1 and state space $\mathcal{X}^r$.
[Hint: The transition probabilities $\hat{w}$ of the new chain are given in terms of the original ones

$$\hat{w}(\vec{x} \to \vec{y}) = \begin{cases} w(x_1, \dots, x_r \to y_r) & \text{if } x_2 = y_1,\, x_3 = y_2, \dots x_r = y_{r-1}\,, \\ 0 & \text{otherwise,} \end{cases} \tag{9.6}$$

where we used the shorthands $\vec{x} \equiv (x_1, \dots, x_r)$ and $\vec{y} = (y_1, \dots, y_r)$].
Figure 9.5 shows the reduction to an order 1 Markov chain in the factor graph language.

What is the content of the global Markov property for Markov chains? Let us start from the case of order 1 chains. Without loss of generality we can choose S as containing one single variable node (let's say the i-th one) while A and B are, respectively the nodes on the left and on the right of i: $A = \{1, \dots, i-1\}$ and $B = \{i+1, \dots, N\}$. The global Markov property reads

$$P(x_1 \dots x_N | x_i) = P(x_1 \dots x_{i-1} | x_i)\, P(x_{i+1} \dots x_N | x_i)\,, \tag{9.7}$$

which is just a rephrasing of the usual Markov condition: $X_{i+1} \dots X_N$ depend upon $X_1 \dots X_i$ uniquely through X_i. We invite the reader to discuss the global Markov property for order-r Markov chains.

Our second example is borrowed from coding theory. Consider the code $\mathfrak{C}$ of block-length $N = 7$ defined by the codebook:

$$\mathfrak{C} = \{(x_1, x_2, x_3, x_4) \in \{0, 1\}^4 \mid x_1 \oplus x_3 \oplus x_5 \oplus x_7 = 0\,, \tag{9.8}$$
$$x_2 \oplus x_3 \oplus x_6 \oplus x_7 = 0\,,\ x_4 \oplus x_5 \oplus x_6 \oplus x_7 = 0\}\,.$$

Let $\mu_0(\underline{x})$ be the uniform probability distribution over the codewords. Then:

$$\mu_0(\underline{x}) = \frac{1}{Z_0}\, \mathbb{I}(x_1 \oplus x_3 \oplus x_5 \oplus x_7 = 0)\, \mathbb{I}(x_2 \oplus x_3 \oplus x_6 \oplus x_7 = 0) \cdot \tag{9.9}$$
$$\cdot\, \mathbb{I}(x_4 \oplus x_5 \oplus x_6 \oplus x_7 = 0)\,,$$

where $Z_0 = 16$ is a normalization constant. This distribution has the form (9.2) and the corresponding factor graph is reproduced in Fig. 9.6.

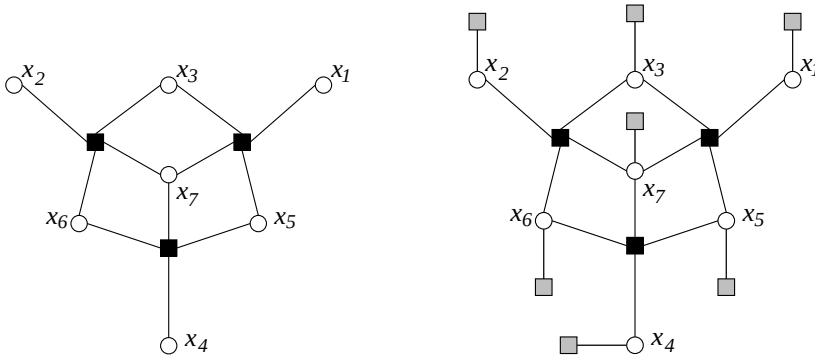

Fig. 9.6 Left: factor graph for the uniform distribution over the code defined in Eq. (9.8). Right: factor graph for the distribution of the transmitted message conditional to the channel output. Gray function nodes encode the information carried by the channel output.

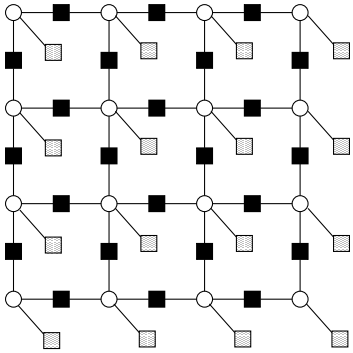

Fig. 9.7 Factor graph for an Edwards-Anderson model with size $L = 4$ in $d = 2$ dimensions. Full squares correspond to pairwise interaction terms $-J_{ij}\sigma_i\sigma_j$. Hatched squares denote magnetic field terms $-B\sigma_i$.

Exercise 9.2 Suppose that a codeword in $\mathfrak{C}$ is transmitted through a binary memoryless channel, and that the message $(y_1, y_2, \dots, y_7)$ is received. As argued in Chap. 6, in order to find the codeword which has been sent, one should consider the probability distribution of the transmitted message conditional to the channel output, cf. Eq. (6.5). Show that the factor graph representation for this distribution is the one given in Fig. 9.6, right-hand frame.

Let us now introduce an example from statistical physics. In Sec. 2.6 we introduced the Edwards-Anderson model, a statistical mechanics model for spin glasses, whose energy function reads: $E(\underline{\sigma}) = -\sum_{(ij)} J_{ij}\sigma_i\sigma_j - B\sum_i \sigma_i$. The Boltzmann distribution can be written as

$$\mu_\beta(\underline{\sigma}) = \frac{1}{Z} \prod_{(ij)} e^{\beta J_{ij}\sigma_i\sigma_j} \prod_i e^{\beta B\sigma_i}\,, \tag{9.10}$$

with i runs over the sites of a d-dimensional cubic lattice of side L: $i \in [L]^d$, and (ij) over the couples of nearest neighbors in the lattice. Once again, this distribution admits a factor graph representation, as shown in Fig. 9.7. This graph includes two types of function nodes. Nodes corresponding to pairwise interaction terms $-J_{ij}\sigma_i\sigma_j$ in the energy function are connected to two neighboring variable nodes. Nodes representing magnetic field terms $-B\sigma_i$ are connected to a unique variable.

The final example comes from combinatorial optimization: Satisfiability is a decision problem introduced in Chap. 3. Given N boolean variables $x_1, \dots, x_N \in \{T, F\}$ and M logical clauses among them, one is asked to find a truth assignment verifying all of the clauses. The logical AND of the M clauses is usually called a formula. Consider for instance the formula over $N = 7$ variables:

$$(x_1 \vee x_2 \vee \overline{x_4}) \wedge (x_2 \vee x_3 \vee x_5) \wedge (\overline{x_4} \vee \overline{x_5}) \wedge (x_5 \vee \overline{x_7} \vee \overline{x_6})\,. \tag{9.11}$$

For a given satisfiability formula, it is quite natural to consider the uniform probability distribution $\mu_{\rm sat}(x_1, \dots, x_N)$ over the truth assignments which satisfy (9.11)(whenever there exists at least one such assignment). A little thought shows that such a distribution can be written in the factorized form (9.2). For instance, the formula (9.11) yields

$$\mu_{\rm sat}(x_1, \dots, x_7) = \frac{1}{Z_{\rm sat}} \mathbb{I}(x_1 \vee x_2 \vee \overline{x_4})\, \mathbb{I}(x_2 \vee x_3 \vee x_5))\, \mathbb{I}(\overline{x_4} \vee \overline{x_5}) \cdot \\ \cdot \mathbb{I}(x_5 \vee \overline{x_7} \vee \overline{x_6})\,, \tag{9.12}$$

where $Z_{\rm sat}$ is the number of distinct truth assignment which satisfy Eq. (9.11). We invite the reader to draw the corresponding factor graph.

Exercise 9.3 Consider the problem of coloring a graph $\mathcal{G}$ with q colors, already encountered in Sec. 3.3. Build a factor graph representation for this problem, and write the associated compatibility functions. [Hint: in the simplest such representation the number of function nodes is equal to the number of edges of $\mathcal{G}$, and every function node has degree 2.]

9.2 Ensembles of factor graphs: definitions

We shall be generically interested in understanding the properties of *ensembles* of probability distributions taking the factorized form (9.2). We introduce here a few useful ensembles of factor graphs. In the simple case where every function node has degree 2, factor graphs are in one to one correspondence with usual graphs, and we are just treating random graph ensembles, as first studied by Erdös and Renyi. The case of arbitrary factor graphs is a simple generalization. From the graph theoretical point of view they can be regarded either as **hyper-graphs** (by associating a vertex

to each variable node and an hyper-edge to each function node), or as bipartite graphs (variable and function nodes are both associated to vertices in this case).

For any integer $k \geq 1$, the **random k-factor graph** with M function nodes and N variables nodes is denoted by $\mathbb{G}_N(k, M)$, and is defined as follows. For each function node $a \in \{1 \dots M\}$, the k-uple ∂a is chosen uniformly at random among the $\binom{N}{k}$ k-uples in $\{1 \dots N\}$.

Sometimes, one may encounter variations of this basic distribution. For instance, it can be useful to prevent any two function nodes to have the same neighborhood, by imposing the condition $\partial a \neq \partial b$ for any $a \neq b$. This can be done in a natural way through the ensemble $\mathbb{G}_N(k, \alpha)$ defined as follows. For each of the $\binom{N}{k}$ k-uples of variables nodes, a function node is added to the factor graph independently with probability $N\alpha/\binom{N}{k}$, and all of the variables in the k-uple are connected to it. The total number M of function nodes in the graph is a random variable, with expectation $M_{\rm av} = \alpha N$.

In the following we shall often be interested in large graphs ($N \to \infty$) with a finite density of function nodes. In $\mathbb{G}_N(k, M)$ this means that $M \to \infty$, with the ratio M/N kept fixed. In $\mathbb{G}_N(k, \alpha)$, the large N limit is taken at α fixed. The exercises below suggests that, for some properties, the distinction between the two graph ensembles does not matter in this limit.

Exercise 9.4 Consider a factor graph from the ensemble $\mathbb{G}_N(k, M)$. What is the probability $p_{\rm dist}$ that for all couples of function nodes, the corresponding neighborhoods are distinct? Show that, in the limit $N \to \infty$, $M \to \infty$ with $M/N \equiv \alpha$ and k fixed

$$p_{\rm dist} = \begin{cases} O(e^{-\frac{1}{2}\alpha^2 N}) & \text{if } k = 1\,, \\ e^{-\alpha^2}[1 + \Theta(N^{-1})] & \text{if } k = 2\,, \\ 1 + \Theta(N^{-k+2}) & \text{if } k \geq 3\,. \end{cases} \tag{9.13}$$

Exercise 9.5 Consider a random factor graph from the ensemble $\mathbb{G}_N(k, \alpha)$, in the large N limit. Show that the probability of getting a number of function nodes M different from its expectation αN by an 'extensive' number (i.e. a number of order N) is exponentially small. In mathematical terms: there exist a constant $A > 0$ such that, for any $\varepsilon > 0$,

$$\mathbb{P}\left[|M - M_{\rm av}| > N\varepsilon\right] \leq 2\, e^{-AN\varepsilon^2}\,. \tag{9.14}$$

Consider the distribution of a $\mathbb{G}_N(k, \alpha)$ random graph conditioned on the number of function nodes being $\overline{M}$. Show that this is the same as the distribution of a $\mathbb{G}_N(k, \overline{M})$ random graph conditioned on all the function nodes having distinct neighborhoods.

An important local property of a factor graph is its **degree profile**. Given a graph, we denote by Λ_i (by P_i) the fraction of variable nodes (function nodes) of degree i. Notice that $\Lambda \equiv \{\Lambda_n : n \geq 0\}$ and $P \equiv \{P_n : n \geq 0\}$ are in fact two distributions over

the non-negative integers (they are both non-negative and normalized). Moreover, they have non-vanishing weight only on a finite number of degrees (at most N for Λ and M for P). The couple (Λ, P) is called the degree profile of the graph F. A practical representation of the degree profile is provided by the generating functions $\Lambda(x) = \sum_{n\geq 0} \Lambda_n\, x^n$ and $P(x) = \sum_{n\geq 0} P_n\, x^n$. Because of the above remarks, both $\Lambda(x)$ and $P(x)$ are in fact finite polynomials with non-negative coefficients. The average variable node (resp. function node) degree is given by $\sum_{n\geq 0} \Lambda_n\, n = \Lambda'(1)$ (resp. $\sum_{n\geq 0} P_n\, n = P'(1)$)

If the graph is randomly generated, its degree profile is a random variable. For instance, in the random k-factor graph ensemble $\mathbb{G}_N(k, M)$ defined above, the variable node degree Λ depends upon the graph realization: we shall investigate some of its properties below. In contrast, its function node profile $P_n = \mathbb{I}(n = k)$ is deterministic.

It is convenient to consider *ensembles* of factor graphs with a prescribed degree profile. We therefore introduce the ensemble of **degree constrained factor graphs** $\mathbb{D}_N(\Lambda, P)$ by endowing the set of graphs with degree profile (Λ, P) with the uniform probability distribution. Notice that the number M of function nodes is fixed by the relation $MP'(1) = N\Lambda'(1)$. A special case which is important in this context is that of **random regular graphs** in which the degrees of variable nodes is fixed, as well as the degree of function nodes. In a (l, k) random regular graph, each variable node has degree l and each function node has degree k, corresponding to $\Lambda(x) = x^l$ and $P(x) = x^k$.

A degree constrained factor graph ensemble is non-empty only if $N\Lambda_n$ and MP_n are integers for any $n \geq 0$. Even if these conditions are satisfied, it is not obvious how to construct efficiently a graph in $\mathbb{D}_N(\Lambda, P)$. Since such ensembles play a crucial role in the theory of sparse graph codes, we postpone this issue to Chap. 11.

9.3 Random factor graphs: basic properties

For the sake of simplicity, we shall study here only the ensemble $\mathbb{G}_N(k, M)$ with $k \geq 2$. Generalizations to graphs in $\mathbb{D}_N(\Lambda, P)$ will be mentioned in Sec. 9.5.1 and further developed in Chap. 11. We study the asymptotic limit of large graphs $N \to \infty$ with k (the degree of function nodes) and $M/N = \alpha$ fixed.

9.3.1 Degree profile

The variable node degree profile $\{\Lambda_n : n \geq 0\}$ is a random variable. By linearity of expectation $\mathbb{E}\,\Lambda_n = \mathbb{P}[\mathsf{deg}_i = n]$, where deg_i is the degree of the node i. Let p be the probability that a uniformly chosen k-uple in $\{1, \dots, N\}$ contains i. It is clear that deg_i is a binomial random variable (defined in Appendix A.3) with parameters M and p. Furthermore, since p does not depend upon the site i, it is equal to the probability that a randomly chosen site belongs to a fixed k-uple. In formulae

$$\mathbb{P}[\mathsf{deg}_i = n] = \binom{M}{n} p^n (1-p)^{M-n}\,, \qquad p = \frac{k}{N}\,. \tag{9.15}$$

If we consider the large graph limit, with n fixed, we get

$$\lim_{N\to\infty} \mathbb{P}\,[\mathsf{deg}_i = n] = \lim_{N\to\infty} \mathbb{E}\,\Lambda_n = e^{-k\alpha}\,\frac{(k\alpha)^n}{n!}\,. \tag{9.16}$$

The degree of site i is asymptotically a Poisson random variable.

How correlated are the degrees of variable nodes? By a simple generalization of the above calculation, we can compute the joint probability distribution of deg_i and deg_j, with $i \neq j$. Think of constructing the graph by choosing a k-uple of variable nodes at a time and adding the corresponding function node to the graph. Each node can have one of four possible 'fates': it connects to both nodes i and j (with probability p_2); it connects only to i or only to j (each case has probability p_1); it connects neither to i nor to j (probability $p_0 \equiv 1-2p_1-p_2$). A little thought shows that $p_2 = k(k-1)/N(N-1)$, $p_1 = k(N-k)/N(N-1)$ and

$$\mathbb{P}[\mathsf{deg}_i = n, \mathsf{deg}_j = m] = \sum_{l=0}^{\min(n,m)} \binom{M}{n-l,\, m-l,\, l} p_2^l p_1^{n+m-2l} p_0^{M-n-m+l} \tag{9.17}$$

where l is the number of function nodes which connect both to i and to j and we used the standard notation for multinomial coefficients (see Appendix A).

Once again, it is illuminating to look at the large graphs limit $N \to \infty$ with n and m fixed. It is clear that the $l = 0$ term dominates the sum (9.17). In fact, the multinomial coefficient is of order $\Theta(N^{n+m-l})$ and the various probabilities are of order $p_0 = \Theta(1)$, $p_1 = \Theta(N^{-1})$, $p_2 = \Theta(N^{-2})$. Therefore the l-th term of the sum is of order $\Theta(N^{-l})$. Elementary calculus then shows that

$$\mathbb{P}[\mathsf{deg}_i = n, \mathsf{deg}_j = m] = \mathbb{P}[\mathsf{deg}_i = n]\, \mathbb{P}[\mathsf{deg}_j = m] + \Theta(N^{-1})\,. \tag{9.18}$$

This shows that, asymptotically, the nodes' degrees are pairwise independent Poisson random variables. This fact can be used to show that the degree profile $\{\Lambda_n : n \geq 0\}$ is, for large graphs, close to its expectation. In fact

$$\begin{aligned} \mathbb{E}\left[(\Lambda_n - \mathbb{E}\Lambda_n)^2\right] &= \frac{1}{N^2} \sum_{i,j=1}^{N} \left\{\mathbb{P}[\mathsf{deg}_i = n, \mathsf{deg}_j = n] - \mathbb{P}[\mathsf{deg}_i = n]\mathbb{P}[\mathsf{deg}_j = n]\right\} \\ &= \Theta(N^{-1})\,, \end{aligned} \tag{9.19}$$

which implies, via Chebyshev inequality, $\mathbb{P}(|\Lambda_n - \mathbb{E}\Lambda_n| \geq \delta\, \mathbb{E}\Lambda_n) = \Theta(N^{-1})$ for any $\delta > 0$.

The pairwise independence expressed in Eq. (9.18) is essentially a consequence of the fact that, given two distinct variable nodes i and j the probability that they are connected to the same function node is of order $\Theta(N^{-1})$. It is easy to see that the same property holds when we consider any finite number of variable nodes. Suppose now that we look at a factor graph from the ensemble $\mathbb{G}_N(k, M)$ conditioned to the function node a being connected to variable nodes $i_1, \dots, i_k$. What is the distribution of the residual degrees $\mathsf{deg}'_{i_1}, \dots, \mathsf{deg}'_{i_k}$ (by residual degree deg'_i, we mean the degree of node i once the function node a has been pruned from the graph)? It is clear that the residual graph is distributed according to the ensemble $\mathbb{G}_N(k, M-1)$. Therefore the residual degrees are (in the large graph limit) independent Poisson random variables with mean $k\alpha$. We can formalize these simple observations as follows.

Proposition 9.4 *Let $i_1, \dots, i_n \in \{1, \dots, N\}$ be n distinct variable nodes, and G a random graph from $\mathbb{G}_N(k, M)$ conditioned to the neighborhoods of m function nodes*

$a_1, \ldots, a_m$ being $\partial a_1, \ldots, \partial a_m$. Denote by deg'_i the degree of variable node i once $a_1, \ldots, a_m$ have been pruned from the graph. In the limit of large graphs $N \to \infty$ with $M/N \equiv \alpha$, k, n and m fixed, the residual degrees $\mathsf{deg}'_{i_1}, \ldots, \mathsf{deg}'_{i_n}$ converge in distribution to independent Poisson random variables with mean $k\alpha$.

This property is particularly useful when investigating the local properties of a $\mathbb{G}_N(k, N\alpha)$ random graph. In particular, it suggests that such local properties are close to the ones of the ensemble $\mathbb{D}_N(\Lambda, P)$, where $P(x) = x^k$ and $\Lambda(x) = \exp[k\alpha(x-1)]$.

A remark: in the above discussion we have focused on the probability of finding a node with some constant degree n in the asymptotic limit $N \to \infty$. One may wonder whether, in a typical graph $G \in \mathbb{G}_N(k, M)$ there may exist some variable nodes with exceptionally large degrees. The exercise below shows that this is not the case.

Exercise 9.6 We want to investigate the typical properties of the maximum variable node degree $\Delta(G)$ in a random graph G from $\mathbb{G}_N(k, M)$.

(*a*) Let $\overline{n}_{\max}$ be the smallest value of $n > k\alpha$ such that $N\mathbb{P}[\mathsf{deg}_i = n] \leq 1$. Show that $\Delta(G) \leq \overline{n}_{\max}$ with probability approaching one in the large graph limit. [Hint: Show that $N\mathbb{P}[\mathsf{deg}_i = \overline{n}_{\max} + 1] \to 0$ at large N]

(*b*) Show that the following asymptotic form holds for $\overline{n}_{\max}$:

$$\frac{\overline{n}_{\max}}{k\alpha e} = \frac{z}{\log(z/\log z)} \left[1 + \Theta\left(\frac{\log\log z}{(\log z)^2}\right)\right], \tag{9.20}$$

where $z \equiv (\log N)/(k\alpha e)$.

(*c*) Let $\underline{n}_{\max}$ be the largest value of n such that $N\mathbb{P}[\mathsf{deg}_i = n] \geq 1$. Show that $\Delta(G) \geq \underline{n}_{\max}$ with probability approaching one in the large graph limit. [Hints: Show that $N\mathbb{P}[\mathsf{deg}_i = \underline{n}_{\max} - 1] \to \infty$ at large N; Apply the second moment method to Z_l, the number of nodes of degree l.]

(*d*) What is the asymptotic behavior of $\underline{n}_{\max}$? How does it compare to $\overline{n}_{\max}$?

9.3.2 Small subgraphs

The next simplest question one may ask, concerning a random graph, is the occurrence in it of a given small subgraph. We shall not give a general treatment of the problem here, but rather work out a few simple examples.

Let's begin by considering a fixed k-uple of variable nodes $i_1, \ldots, i_k$ and ask for the probability p that they are connected by a function node in a graph $G \in \mathbb{G}_N(k, M)$. In fact, it is easier to compute the probability that they are *not* connected:

$$1 - p = \left[1 - \binom{N}{k}^{-1}\right]^M . \tag{9.21}$$

The quantity in brackets is the probability that a given function node *is not* a neighbor of $i_1, \ldots, i_k$. It is raised to the power M because the M function nodes are independent

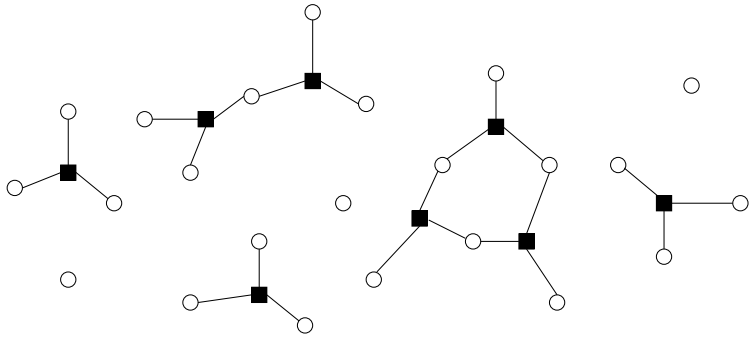

Fig. 9.8 A factor graph from the $\mathbb{G}_N(k, M)$ with $k = 3$, $N = 23$ and $M = 8$. It contains $Z_{\rm isol} = 3$ isolated function nodes, $Z_{\rm isol,2} = 1$ isolated couples of function nodes and $Z_{\rm cycle,3} = 1$ cycle of length 3. The remaining 3 variable nodes have degree 0.

in the model $\mathbb{G}_N(k, M)$. In the large graph limit, we get

$$p = \frac{\alpha\, k!}{N^{k-1}}[1 + \Theta(N^{-1})]\,. \tag{9.22}$$

This confirms an observation of the previous section: for any fixed set of nodes, the probability that a function node connects any two of them vanishes in the large graph limit.

As a first example, let's ask how many isolated function nodes appear in a graph $G \in \mathbb{G}_N(k, M)$. We say that a node is isolated if all the neighboring variable nodes have degree one. Call the number of such function nodes $Z_{\rm isol}$. It is easy to compute the expectation of this quantity

$$\mathbb{E}\, Z_{\rm isol} = M \left[\binom{N}{k}^{-1} \binom{N-k}{k} \right]^{M-1} . \tag{9.23}$$

The factor M is due to the fact that each of the M function nodes can be isolated. Consider one such node a and its neighbors $i_1, \dots, i_k$. The factor in $\binom{N}{k}^{-1}\binom{N-k}{k}$ is the probability that a function node $b \neq a$ is not incident on any of the variables $i_1, \dots, i_k$. This must be counted for any $b \neq a$, hence the exponent $M - 1$. Once again, things become more transparent in the large graph limit:

$$\mathbb{E}\, Z_{\rm isol} = N\alpha e^{-k^2\alpha}[1 + \Theta(N^{-1})]\,. \tag{9.24}$$

So, there is a non-vanishing density of isolated function nodes, $\mathbb{E}\, Z_{\rm isol}/N$. This density approaches 0 at small α (because there are few function nodes) and at large α (because function nodes are unlikely to be isolated). A more refined analysis shows that indeed $Z_{\rm isol}$ is tightly concentrated around its expectation: the probability of an order N fluctuation vanishes exponentially as $N \to \infty$.

There is a way of getting the asymptotic behavior (9.24) without going through the exact formula (9.23). We notice that $\mathbb{E}\, Z_{\rm isol}$ is equal to the number of function nodes

($M = N\alpha$) times the probability that the neighboring variable nodes $i_1, \dots, i_k$ have degree 0 in the residual graph. Because of Proposition 9.4, the degrees $\mathsf{deg}'_{i_1}, \dots, \mathsf{deg}'_{i_k}$ are approximatively i.i.d. Poisson random variables with mean $k\alpha$. Therefore the probability for all of them to vanish is close to $(e^{-k\alpha})^k = e^{-k^2\alpha}$.

Of course this last type of argument becomes extremely convenient when considering small structures which involve more than one function node. As a second example, let us compute the number $Z_{\rm isol,2}$ of couples of function nodes which have exactly one variable node in common and are isolated from the rest of the factor graph (for instance in the graph of Fig. 9.8, we have $Z_{\rm isol,2} = 1$). One gets

$$\mathbb{E}\, Z_{\rm isol,2} = \binom{N}{2k-1} \cdot \frac{k}{2} \binom{2k-1}{k} \cdot \left(\frac{\alpha k!}{N^{k-1}}\right)^2 \cdot (e^{-k\alpha})^{2k-1} \left[1 + \Theta\left(\frac{1}{N}\right)\right] . \quad (9.25)$$

The first factor counts the ways of choosing the $2k-1$ variable nodes which support the structure. Then we count the number of way of connecting two function nodes to $(2k-1)$ variable nodes in such a way that they have only one variable in common. The third factor is the probability that the two function nodes are indeed present (see Eq. (9.22)). Finally we have to require that the residual graph of all the $(2k-1)$ variable nodes is 0, which gives the factor $(e^{-k\alpha})^{2k-1}$. The above expression is easily rewritten as

$$\mathbb{E}\, Z_{\rm isol,2} = N \cdot \frac{1}{2}(k\alpha)^2\, e^{-k(2k-1)\alpha} \left[1 + \Theta(1/N)\right] . \quad (9.26)$$

With some more work one can prove again that $Z_{\rm isol,2}$ is in fact concentrated around its expected value: a random factor graph contains a finite density of isolated couples of function nodes.

Let us consider, in general, the number of small subgraphs of some definite type. Its most important property is how it scales with N in the large N limit. This is easily found. For instance let's have another look at Eq. (9.25): N enters only in counting the $(2k-1)$-uples of variable nodes which can support the chosen structure, and in the probability of having two function nodes in the desired positions. In general, if we consider a small subgraph with v variable nodes and f function nodes, the number $Z_{v,f}$ of such structures has an expectation which scales as:

$$\mathbb{E}\, Z_{v,f} \sim N^{v-(k-1)f} . \quad (9.27)$$

This scaling has important consequences on the nature of small structures which appear in a large random graph. For discussing such structures, it is useful to introduce the notions of 'connected (sub-)graph', of 'tree', of 'path' in a factor graph exactly in the same way as in usual graphs, whereby both variable and function nodes are viewed as vertices (see Chap. 3). We further define a **component** of the factor graph G as a subgraph C which is is connected and isolated, in the sense that there is no path between a node of C and a node of $G \backslash C$

Consider a connected factor graph with v variable nodes and f function nodes, all of them having degree k. This graph is a tree if and only if $v = (k-1)f+1$. Call $Z_{{\rm tree},v}$ the number of isolated trees over v variable nodes which are contained in a $\mathbb{G}_N(k, M)$

random graph. Because of Eq. (9.27), we have $\mathbb{E}\, Z_{\mathrm{tree},v} \sim N$: a random graph contains a finite density (when $N \to \infty$) of trees of any finite size. On the other hand, connected subgraphs which are not trees must have $v < (k-1)f + 1$, and Eq. (9.27) shows that their number does not grow with N. In other words, most (more precisely, all but a vanishing fraction) of *finite* components of a random factor graph are trees.

Exercise 9.7 Consider the largest component in the graph of Fig. 9.8 (the one with three function nodes), and let $Z_{\mathrm{cycle},3}$ be the number of times it occurs as a component of a $\mathbb{G}_N(k, M)$ random graph. Compute $\mathbb{E}\, Z_{\mathrm{cycle},3}$ in the large graph limit.

Exercise 9.8 A factor graph is said to be **unicyclic** if it contains a unique (up to shifts) closed, self-avoiding path $\omega_0, \omega_1, \dots, \omega_\ell = \omega_0$ ('self-avoiding' means that for any $t, s \in \{0 \dots \ell - 1\}$ with $t \neq s$, one has $\omega_t \neq \omega_s$).

(*a*) Show that a connected factor graph with v variable nodes and f function nodes, all of them having degree k is unicyclic if and only if $v = (k-1)f$.

(*b*) Let $Z_{\mathrm{cycle},v}(N)$ be the number of unicyclic components over v nodes in a $\mathbb{G}_N(k, M)$ random graph. Use Eq. (9.27) to show that $Z_{\mathrm{cycle},v}$ is finite with high probability in the large graph limit. More precisely, show that $\lim_{n\to\infty} \lim_{N\to\infty} \mathbb{P}_{\mathbb{G}_N}[Z_{\mathrm{cycle},v} \geq n] = 0$.

9.4 Random factor graphs: The giant component

We have just argued that most finite size components of a $\mathbb{G}_N(k, \alpha N)$ factor graph are trees in the large N limit. However, finite size tree do not always exhaust the graph. It turns out that when α becomes larger than a threshold value, a 'giant component' appears in the graph. This is a connected component containing an extensive (proportional to N) number of variable nodes, with many cycles.

9.4.1 Nodes in finite trees

We want to estimate which fraction of a random graph from the $\mathbb{G}_N(k, \alpha N)$ ensemble is covered by finite size trees. This fraction is defined as:

$$x_{\mathrm{tr}}(\alpha, k) \equiv \lim_{s\to\infty} \lim_{N\to\infty} \frac{1}{N}\, \mathbb{E}\, N_{\mathrm{trees},s}\,, \tag{9.28}$$

where $N_{\mathrm{trees},s}$ is the number of sites contained in trees of size not larger than s. In order to compute $\mathbb{E}\, N_{\mathrm{trees},s}$, we use the number of trees of size equal to s, which we denote by $Z_{\mathrm{trees},s}$. Using the approach discussed in the previous Section, we get

$$\mathbb{E}\, N_{\mathrm{trees},s} = \sum_{v=0}^{s} v \cdot \mathbb{E}\, Z_{\mathrm{trees},v} = \tag{9.29}$$
$$= \sum_{v=0}^{s} v \binom{N}{v} \cdot T_k(v) \cdot \left(\frac{\alpha k!}{N^{k-1}}\right)^{\frac{v-1}{k-1}} \cdot (e^{-k\alpha})^v \left[1 + \Theta\left(\frac{1}{N}\right)\right] =$$
$$= N(\alpha k!)^{-1/(k-1)} \sum_{v=0}^{s} \frac{1}{(v-1)!} T_k(v) \left[(\alpha k!)^{\frac{1}{k-1}} e^{-k\alpha}\right]^v + \Theta(1)\,,$$

where $T_k(v)$ is the number of trees which can be built out of v distinct variable nodes and $f = (v-1)/(k-1)$ function nodes of degree k. The computation of $T_k(v)$ is a classical piece of enumerative combinatorics which is developed in Sec. 9.4.3 below. The result is

$$T_k(v) = \frac{(v-1)!\, v^{f-1}}{(k-1)!^f\, f!}\,, \tag{9.30}$$

and the generating function $\widehat{T}_k(z) = \sum_{v=1}^{\infty} T_k(v) z^v/(v-1)!$, which we need in order to compute $\mathbb{E}N_{\mathrm{trees,s}}$ from (9.29), is found to satisfy the self consistency equation:

$$\widehat{T}_k(z) = z \, \exp\left\{\frac{\widehat{T}_k(z)^{k-1}}{(k-1)!}\right\}. \tag{9.31}$$

It is a simple exercise to see that, for any $z \geq 0$, this equation has two solutions such that $\widehat{T}_k(z) \geq 0$, the relevant one being the smallest of the two (this is a consequence of the fact that $\widehat{T}_k(z)$ has a regular Taylor expansion around $z = 0$). Using this characterization of $\widehat{T}_k(z)$, one can show that $x_{\mathrm{tr}}(\alpha, k)$ is the smallest positive solution of the equation

$$x_{\mathrm{tr}} = \exp\left(-k\alpha + k\alpha\, x_{\mathrm{tr}}^{k-1}\right). \tag{9.32}$$

This equation is solved graphically in Fig. 9.9, left frame. In the range $\alpha \leq \alpha_{\mathrm{p}} \equiv 1/(k(k-1))$, the only non-negative solution is $x_{\mathrm{tr}} = 1$: all but a vanishing fraction of nodes belong to finite size trees. When $\alpha > \alpha_{\mathrm{p}}$, the solution has $0 < x_{\mathrm{tr}} < 1$: the fraction of nodes in finite trees is strictly smaller than one.

9.4.2 Size of the giant component

This result is somewhat surprising. For $\alpha > \alpha_{\mathrm{p}}$, a strictly positive fraction of variable nodes does not belong to any finite tree. On the other hand, we saw in the previous Section that finite components with cycles contain a vanishing fraction of nodes. Where are the other $N(1 - x_{\mathrm{tr}})$ nodes? It turns out that, roughly speaking, they belong to a unique connected component, the so-called giant component, which is not a tree. One basic result describing this phenomenon is the following.

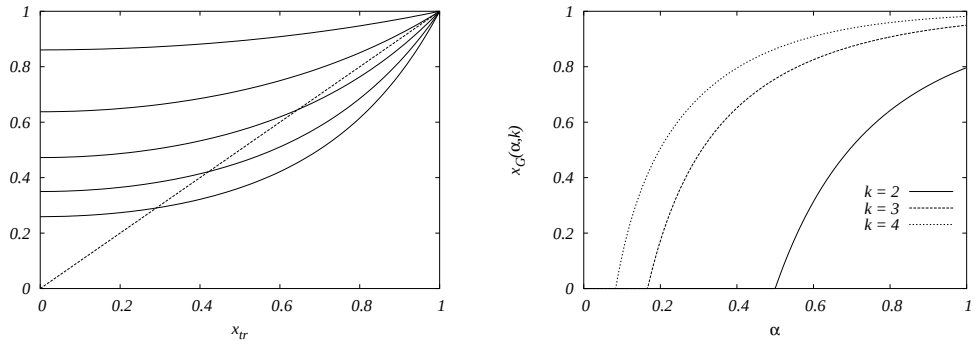

Fig. 9.9 Left: graphical representation of Eq. (9.32) for the fraction of nodes of a $\mathbb{G}_N(k, M)$ random factor graph that belong to finite-size tree components. The curves refer to $k = 3$ and (from top to bottom) $\alpha = 0.05, 0.15, 0.25, 0.35, 0.45$. Right: typical size of the giant component.

Theorem 9.5 *Let X_1 be the size of the largest connected component in a $\mathbb{G}_N(k, M)$ random graph with $M = N[\alpha + o_N(1)]$, and $x_{\rm G}(\alpha, k) = 1 - x_{\rm tr}(\alpha, k)$ where $x_{\rm tr}(\alpha, k)$ is defined as the smallest solution of (9.32). Then, for any positive ε,*

$$|X_1 - N x_{\rm G}(\alpha, k)| \le N\varepsilon\,, \tag{9.33}$$

with high probability.

Furthermore, the giant component contains many loops. Let us define the **cyclic number** c of a factor graph containing v vertices and f function nodes of degree k, as $c = v - (k-1)f - 1$. Then the cyclic number of the giant component is $c = \Theta(N)$ with high probability.

Exercise 9.9 Convince yourself that there cannot be more than one component of size $\Theta(N)$. Here is a possible route. Consider the event of having two connected components of sizes $\lfloor Ns_1 \rfloor$ and $\lfloor Ns_2 \rfloor$ for two fixed positive numbers s_1 and s_2 in a $\mathbb{G}_N(k, M)$ random graph with $M = N[\alpha + o_N(1)]$ (with $\alpha \ge s_1 + s_2$). In order to estimate the probability of such an event, imagine constructing the $\mathbb{G}_N(k, M)$ graph by adding one function node at a time. Which condition must hold when the number of function nodes is $M - \Delta M$? What can happen to the last ΔM nodes? Now take $\Delta M = \lfloor N^\delta \rfloor$ with $0 < \delta < 1$.

The appearance of a giant component is sometimes referred to as **percolation on the complete graph** and is one of the simplest instance of a phase transition. We shall now give a simple heuristic argument which predicts correctly the typical size of the giant component. This argument can be seen as the simplest example of the 'cavity method' that we will develop in the next Chapters. We first notice that, by linearity of expectation, $\mathbb{E}\, X_1 = N x_{\rm G}$, where $x_{\rm G}$ is the probability that a given variable node i belongs to the giant component. In the large graph limit, site i is connected to $l(k-1)$ distinct variable nodes, l being a Poisson random variable of mean $k\alpha$ (see Sec. 9.3.1).

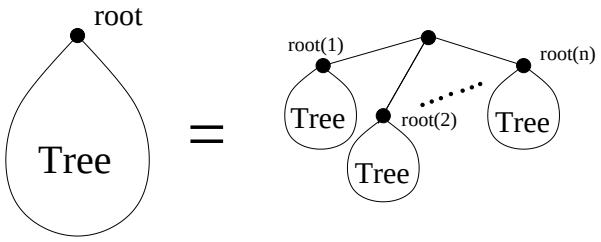

Fig. 9.10 A rooted tree G on $v+1$ vertices can be decomposed into a root and the union of n rooted trees $G_1, \dots, G_n$, respectively on $v_1, \dots, v_n$ vertices.

The node i belongs to the giant component if any of its $l(k-1)$ neighbors does. If we assume that the $l(k-1)$ neighbors belong to the giant component independently with probability $x_{\rm G}$, then we get

$$x_{\rm G} = \mathbb{E}_l[1-(1-x_{\rm G})^{l(k-1)}]\,. \tag{9.34}$$

where l is Poisson distributed with mean $k\alpha$. Taking the expectation, we get

$$x_{\rm G} = 1-\exp[-k\alpha + k\alpha(1-x_{\rm G})^{k-1}]\,, \tag{9.35}$$

which coincides with Eq. (9.32) if we set $x_{\rm G} = 1-x_{\rm tr}$.

The above argument has several flaws but only one of them is serious. In writing Eq. (9.34), we assumed that the probability that none of l randomly chosen variable nodes belongs to the giant component is just the product of the probabilities that each of them does not. In the present case it is not difficult to fix the problem, but in subsequent Chapters we shall see several examples of the same type of heuristic reasoning where the solution is less straightforward.

9.4.3 Appendix: counting trees

This section is a technical appendix devoted to the computation $T_k(v)$, the number of trees with v variable nodes, when function nodes have degree k. Let us begin by considering the case $k=2$. Notice that, if $k=2$, we can uniquely associate to any factor graph F an ordinary graph G obtained by replacing each function node by an edge joining the neighboring variables (for basic definitions on graphs we refer to Chap. 3). In principle G may contain multiple edges but this does not concern us as long as we stick to F being a tree. Therefore $T_2(v)$ is just the number of ordinary (non-factor) trees on v distinct vertices. Rather than computing $T_2(v)$ we shall compute the number $T_2^*(v)$ of **rooted** trees on v distinct vertices. Recall that a rooted graph is just a couple (G, i_*) where G is a graph and i_* is a distinguished node in G. Of course we have the relation $T_2^*(v) = vT_2(v)$.

Consider now a rooted tree on $v+1$ vertices, and assume that the root has degree n (of course $1 \le n \le v$). Erase the root together with its edges and mark the n vertices that were connected to the root. One is left with n rooted trees of sizes $v_1, \dots, v_n$ such that $v_1 + \cdots + v_n = v$. This naturally leads to the recursion

$$T_2^*(v+1) = (v+1)\sum_{n=1}^{v}\frac{1}{n!}\sum_{\substack{v_1\dots v_n>0\\ v_1+\cdots+v_n=v}}\binom{v}{v_1,\cdots,v_n}\,T_2^*(v_1)\cdots T_2^*(v_n)\,, \qquad (9.36)$$

which holds for any $v \geq 1$. Together with the initial condition $T_2^*(1) = 1$, this relation allows to determine recursively $T_2^*(v)$ for any $v > 0$. This recursion is depicted in Fig. 9.10.

The recursion is most easily solved by introducing the generating function $\widehat{T}(z) = \sum_{v>0} T_2^*(v)\, z^v/v!$. Using this definition in Eq. (9.36), we get

$$\widehat{T}(z) = z\,\exp\{\widehat{T}(z)\}\,, \qquad (9.37)$$

which is closely related to the definition of Lambert's W function (usually written as $W(z)\exp(W(z)) = z$). One has in fact the identity $\widehat{T}(z) = -W(-z)$. The expansion of $\widehat{T}(z)$ in powers of z can be obtained through Lagrange's inversion method (see Exercise below). We get $T_2^*(v) = v^{v-1}$, and therefore $T_2(v) = v^{v-2}$. This result is known as **Cayley formula** and is one of the most famous results in enumerative combinatorics.

Exercise 9.10 Assume that the generating function $A(z) = \sum_{n>0} A_n z^n$ is solution of the equation $z = f(A(z))$, with f an analytic function such that $f(0) = 0$ and $f'(0) = 1$. Use Cauchy formula $A_n = \oint \frac{dz}{2\pi i}\, z^{-n-1}\, A(z)$ to show that

$$A_n = \mathsf{coeff}\left\{f'(x)\,(x/f(x))^{n+1};\, x^{n-1}\right\}\,. \qquad (9.38)$$

Use this result, known as 'Lagrange inversion method', to compute the power expansion of $\widehat{T}(z)$ and prove Cayley formula $T_2(v) = v^{v-2}$.

Let us now return to the generic k case. The reasoning is similar to the $k = 2$ case. One finds after some work that the generating function $\widehat{T}_k(z) \equiv \sum_{v>0} T_k^*(v) z^v/v!$ satisfies the equation:

$$\widehat{T}_k(z) = z\,\exp\left\{\frac{\widehat{T}_k(z)^{k-1}}{(k-1)!}\right\}\,, \qquad (9.39)$$

from which one deduces the number of trees with v variable nodes:

$$T_k^*(v) = \frac{v!\,v^{f-1}}{(k-1)!^f\, f!}\,. \qquad (9.40)$$

In this expression the number of function nodes f is fixed by $v = (k-1)f + 1$.

9.5 The locally tree-like structure of random graphs

9.5.1 Neighborhood of a node

There exists a natural notion of distance between variable nodes of a factor graph. Given a path $(\omega_0, \dots, \omega_\ell)$ on the factor graph, we define its length as the number of

function nodes in it. Then the **distance** between two variable nodes is defined as the length of the shortest path connecting them (by convention it is set to $+\infty$ when the nodes belong to distinct connected components). We also define the **neighborhood** of radius r of a variable node i, denoted by $\mathsf{B}_{i,r}(F)$ as the subgraph of F including all the variable nodes at distance at most r from i, and all the function nodes connected only to these variable nodes.

What does the neighborhood of a typical node look like in a random graph? It is convenient to step back for a moment from the $\mathbb{G}_N(k,M)$ ensemble and consider a degree-constrained factor graph $F \stackrel{\mathrm{d}}{=} \mathbb{D}_N(\Lambda,P)$. We furthermore define the **edge perspective** degree profiles as $\lambda(x) \equiv \Lambda'(x)/\Lambda'(1)$ and $\rho(x) \equiv P'(x)/P'(1)$. These are polynomials

$$\lambda(x) = \sum_{l=1}^{l_{\max}} \lambda_l \, x^{l-1} \,, \qquad \rho(x) = \sum_{k=1}^{k_{\max}} \rho_k \, x^{k-1} \,, \tag{9.41}$$

where λ_l (respectively ρ_k) is the probability that a randomly chosen edge in the graph is adjacent to a variable node (resp. function node) of degree l (degree k). The explicit formulae

$$\lambda_l = \frac{l\Lambda_l}{\sum_{l'} l'\Lambda_{l'}} \,, \qquad \rho_k = \frac{kP_k}{\sum_{k'} k'P_{k'}} \,, \tag{9.42}$$

are derived by noticing that the graph F contains $nl\Lambda_l$ (resp. mkP_k) edges adjacent to variable nodes of degree l (resp. function nodes of degree k).

Imagine constructing the neighborhoods of a node i of increasing radius r. Given $\mathsf{B}_{i,r}(F)$, let $i_1,\dots,i_L$ be the nodes at distance r from i, and $\mathsf{deg}'_{i_1},\dots,\mathsf{deg}'_{i_L}$ their degrees in the residual graph $F\setminus\mathsf{B}_{i,r}(F)$. Arguments analogous to the ones leading to Proposition 9.4 imply that $\mathsf{deg}'_{i_1},\dots,\mathsf{deg}'_{i_L}$ are asymptotically i.i.d. random variables with $\mathsf{deg}'_{i_n} = l_n - 1$, and l_n distributed according to λ_{l_n}. An analogous result holds for function nodes (just invert the roles of variable and function nodes).

This motivates the following definition of an r-generations tree ensemble $\mathbb{T}_r(\Lambda,P)$. If $r=0$ there is a unique element in the ensemble: a single isolated node, which is attributed the generation number 0. If $r>0$, first generate a tree from the $\mathbb{T}_{r-1}(\Lambda,P)$ ensemble. Then for each variable-node i of generation $r-1$ draw an independent integer $l_i \geq 1$ distributed according to λ_l and add to the graph $l_i - 1$ function nodes connected to the variable i (unless $r=1$, in which case l_i function nodes are added, with l_i distributed according to Λ_{l_i}). Next, for each of the newly added function nodes $\{a\}$, draw an independent integer $k_a \geq 1$ distributed according to ρ_k and add to the graph $k_a - 1$ variable nodes connected to the function a. Finally, the new variable nodes are attributed the generation number r. The case of uniformly chosen random graphs where function nodes have a fixed degree, k, corresponds to the tree ensemble $\mathbb{T}_r(e^{k\alpha(x-1)},x^k)$. In this case, it is easy to check that the degrees in the residual graph have a Poisson distribution with mean $k\alpha$, in agreement with proposition 9.4. With a slight abuse of notation, we shall use the shorthand $\mathbb{T}_r(k,\alpha)$ to denote this tree ensemble.

It is not unexpected that $\mathbb{T}_r(\Lambda, P)$ constitutes a good model for r-neighborhoods in the degree-constrained ensemble. Analogously, $\mathbb{T}_r(k, \alpha)$ is a good model for r-neighborhoods in the $\mathbb{G}_N(k, M)$ ensemble when $M \simeq N\alpha$. This is made more precise below.

Theorem 9.6 *Let F be a random factor graph in the $\mathbb{D}_N(\Lambda, P)$ ensemble (respectively in the $\mathbb{G}_N(k, M)$ ensemble), let i be a uniformly random variable node in F, and r a non-negative integer. Then $\mathsf{B}_{i,r}(F)$ converges in distribution to $\mathbb{T}_r(\Lambda, P)$ (resp. to $\mathbb{T}_r(k, \alpha)$) as $N \to \infty$ with Λ, P fixed (α, k fixed).*

In other words, the factor graph F looks locally like a random tree from the ensemble $\mathbb{T}_r(\Lambda, P)$.

9.5.2 Loops

We have seen that in the large graph limit, a factor graph $F \stackrel{\mathrm{d}}{=} \mathbb{G}_N(k, M)$ converges locally to a tree. Furthermore, it has been shown in Sec. 9.3.2 that the number of 'small' cycles in such a graph is only $\Theta(1)$ an $N \to \infty$. It is therefore natural to wonder at which distance from any given node loops start playing a role.

More precisely, let i be a uniformly random node in F. We would like to know what is the typical length of the shortest loop through i. Of course, this question has a trivial answer if $k(k-1)\alpha < 1$, since in this case most of the variable nodes belong to small tree components, cf. Sec. 9.4. We shall hereafter consider $k(k-1)\alpha > 1$.

A heuristic guess of the size of this loop can be obtained as follows. Assume that the neighborhood $\mathsf{B}_{i,r}(F)$ is a tree. Each function node has $k-1$ adjacent variable nodes at the successive generation. Each variable node has a Poisson number adjacent function nodes at the successive generation, with mean $k\alpha$. Therefore the average number of variable nodes at a given generation is $[k(k-1)\alpha]$ times the number at the previous generation. The total number of nodes in $\mathsf{B}_{i,r}(F)$ is about $[k(k-1)\alpha]^r$, and loops will appear when this quantity becomes comparable with the total number of nodes in the graph. This yields $[k(k-1)\alpha]^r = \Theta(N)$, or $r = \log N / \log[k(k-1)\alpha]$. This is of course a very crude argument, but it is also a very robust one: one can for instance change N with $N^{1\pm\varepsilon}$ affecting uniquely the prefactor. It turns out that the result is correct, and can be generalized to the $\mathbb{D}_N(\Lambda, P)$ ensemble:

Proposition 9.7 *Let F be a random factor graph in the $\mathbb{D}_N(\Lambda, P)$ ensemble (in the $\mathbb{G}_N(k, M)$ ensemble), let i be a uniformly chosen random variable node in F, and ℓ_i the length of the shortest loop in F through i. Assume that $c = \lambda'(1)\rho'(1) > 1$ ($c = k(k-1)\alpha > 1$). Then, with high probability,*

$$\ell_i = \frac{\log N}{\log c}[1 + o(1)]\,. \tag{9.43}$$

We shall refer the reader to the literature for the proof, the following exercise gives a slightly more precise, but still heuristic, version of the previous argument.

Exercise 9.11 Assume that the neighborhood $\mathsf{B}_{i,r}(F)$ is a tree and that it includes n 'internal' variables nodes (i.e. nodes whose distance from i is smaller than r), $n_{\rm l}$ 'boundary' variable nodes (whose distance from i is equal to r), and m function nodes. Let F_r be the residual graph, i.e. F minus the subgraph $\mathsf{B}_{i,r}(F)$. It is clear that $F_r \stackrel{\rm d}{=} \mathbb{G}_{N-n}(k, M-m)$. Show that the probability, p_r, that a function node of F_r connects two of the variable nodes on the boundary of $\mathsf{B}_{i,r}(F)$ is

$$p_r = 1 - \left[(1-q)^k + k\,(1-q)^{k-1}\,q\right]^{M-m}\,, \tag{9.44}$$

where $q \equiv n_{\rm l}/(N-n)$. As a first estimate of p_r, we can substitute in this expression $n_{\rm l}$, n, m, with their expectations (in the tree ensemble) and call $\overline{p}_r$ the corresponding estimate. Assuming that $r = \rho \frac{\log N}{\log[k(k-1)\alpha]}$, show that

$$\overline{p}_r = 1 - \exp\left\{-\frac{1}{2}k(k-1)\alpha N^{2\rho-1}\right\}\left[1 + O(N^{-2+3\rho})\right]. \tag{9.45}$$

If $\rho > 1/2$, this indicates that, under the assumption that there is no loop of length $2r$ or smaller through i, there is, with high probability, a loop of length $2r+1$. If, on the other hand, $\rho < 1/2$, it indicates that there is no loop of length $2r+1$ *or smaller* through i. This argument suggests that the length of the shortest loop through i is about $\frac{\log N}{\log[k(k-1)\alpha]}$.

Notes

A nice introduction to factor graphs is the paper (Kschischang, Frey and Loeliger, 2001), see also (Aji and McEliece, 2000). They are related to graphical models (Jordan, 1998), to Bayesian networks (Pearl, 1988), and to Tanner graphs in coding (Tanner, 1981). Among the alternatives to factor graphs, it is worth recalling 'normal realizations' discussed by Forney in (Forney, 2001).

The proof of the Hammersley-Clifford theorem (initially motivated by the probabilistic modeling of some physical problems) goes back to 1971. A proof, more detailed references and some historical comments can be found in (Clifford, 1990).

The theory of random graphs has been pioneered by Erdös and Renyi (Erdös and Rényi, 1960). The emergence of a giant component in a random graph is a classic result which goes back to their work. Two standard textbooks on random graphs like (Bollobás, 2001) and (Janson, Luczak and Ruciński, 2000) provide in particular a detailed study of the phase transition. Graphs with constrained degree profiles were studied in (Bender and Canfield, 1978). A convenient 'configuration mode' for analyzing them was introduced in (Bollobás, 1980) and allowed for the location of the phase transition in (Molloy and Reed, 1995). Finally, (Wormald, 1999) provides a useful survey (including short loop properties) of degree constrained ensembles.

For general background on hyper-graphs, see (Duchet, 1995). The threshold for the emergence of a giant component in a random hyper-graph with edges of fixed size k (corresponding to the factor graph ensemble $\mathbb{G}_N(k, M)$) is discussed in (Schmidt-

Pruzan and Shamir, 1985). The neighborhood of the threshold is analyzed in (Karoński and Luczak, 2002) and references therein.

In enumerating trees we used generating functions. This aproach to combinatorics is developed thoroughly in (Flajolet and Sedgewick, 2008).

Ensembles with hyper-edges of different sizes were considered recently in combinatorics (Darling and Norris, 2005), as well as in coding theory (as code ensembles). Our definitions and notations for degree profiles and degree constrained ensembles follows the coding literature (Luby, Mitzenmacher, Shokrollahi, Spielman and Stemann, 1997; Richardson and Urbanke, 2001*a*).

The local structure of random graphs, and of more complex random objects (in particular random *labeled* graphs) is the object of the theory of *local weak convergence* (Aldous and Steele, 2003).

10
Satisfiability

Because of Cook's theorem, see Chapter 3, satisfiability lies at the heart of computational complexity theory: this fact has motivated an intense research activity on this problem. This Chapter will not be a comprehensive introduction to such a vast topic, but rather present some selected research directions. In particular, we shall pay special attention to the definition and analysis of ensembles of random satisfiability instances. There are various motivations for studying random instances. In order to test and improve algorithms that aim at solving satisfiability, it is highly desirable to have an automatic generator of 'hard' instances at hand. As we shall see, properly 'tuned' ensembles provide such a generator. Also, the analysis of ensembles has revealed a rich structure and stimulated fruitful contacts with other disciplines. The present chapter focuses on 'standard' algorithmic and probabilistic approaches. We shall come back to satisfiability, using methods inspired from statistical physics, in Ch. **??**.

Sec. 10.1 recalls the definition of satisfiability and introduces some standard terminology. A basic, and widely adopted, strategy for solving decision problems consists in exploring exhaustively the tree of possible assignments of the problem's variables. Sec. 10.2 presents a simple implementation of this strategy. In Sec. 10.3 we introduce some important ensembles of random instances. The hardness of satisfiability depends on the maximum clause length. When clauses have length 2, the decision problem is solvable in polynomial time. This is the topic of Sec. 10.4. Finally, in Sec. 10.5 we discuss the existence of a phase transition for random K-satisfiability with $K \geq 3$, when the density of clauses is varied, and derive some rigorous bounds on the location of this transition.

10.1 The satisfiability problem

10.1.1 SAT and UNSAT formulas

An instance of the satisfiability problem is defined in terms of N Boolean variables, and a set of M constraints between them, where each constraint takes the special form of a clause. A clause is the logical `OR` of some variables or their negations. Here we shall adopt the following representation: a variable x_i, with $i \in \{1, \dots, N\}$, takes values in $\{0, 1\}$, 1 corresponding to 'true', and 0 to 'false'; the negation of x_i is $\overline{x}_i \equiv 1 - x_i$. A variable or its negation is called a literal, and we shall denote it by z_i, with $i \in \{1, \dots, N\}$ (therefore z_i denotes any of x_i, $\overline{x}_i$). A clause a, with $a \in \{1, ..., M\}$, involving K_a variables is a constraint which forbids exactly one among the 2^{K_a} possible assignments to these K_a variables. It is written as the logical `OR` (denoted by $\vee$) function of some variables or their negations. For instance the clause $x_2 \vee \overline{x}_{12} \vee x_{37} \vee \overline{x}_{41}$

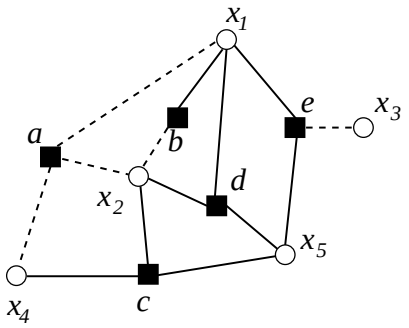

Fig. 10.1 Factor graph representation of the formula $(\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_4) \wedge (x_1 \vee \overline{x}_2) \wedge (x_2 \vee x_4 \vee x_5) \wedge (x_1 \vee x_2 \vee x_5) \wedge (x_1 \vee \overline{x}_3 \vee x_5)$.

is satisfied by all the variables' assignments except those where $x_2 = 0, x_{12} = 1, x_{37} = 0, x_{41} = 1$. When it is not satisfied, a clause is said to be violated.

We denote by ∂a the subset $\{i_1^a, \dots, i_{K_a}^a\} \subseteq \{1, \dots, N\}$ containing the indices of the $K_a = |\partial a|$ variables involved in clause a. Then clause a is written as $C_a = z_{i_1^a} \vee z_{i_2^a} \vee \cdots \vee z_{i_{K_a}^a}$. An instance of the satisfiability problem can be summarized as the logical formula (in the so-called **conjunctive normal form (CNF)**):

$$F = C_1 \wedge C_2 \wedge \cdots \wedge C_M \,. \tag{10.1}$$

As we have seen in Sec. 9.1.2, there exists [1] a simple and natural representation of a satisfiability formula as a factor graph associated with the indicator function $\mathbb{I}(\underline{x} \text{ satisfies } F)$. Actually, it is often useful to use a slightly more elaborate factor graph with two types of edges: A full edge is drawn between a variable vertex i and a clause vertex a whenever x_i appears in a, and a dashed edge is drawn whenever $\overline{x}_i$ appears in a. In this way there is a one to one correspondence between a CNF formula and its graph. An example is shown in Fig. 10.1.

Given the formula F, the question is whether there exists an assignment of the variables x_i to $\{0, 1\}$ (among the 2^N possible assignments), such that the formula F is true. An algorithm solving the satisfiability problem must be able, given a formula F, to either answer 'YES' (the formula is then said to be **SAT**), and provide such an assignment, called a **SAT-assignment**, or to answer 'NO', in which case the formula is called **UNSAT**. The restriction of the satisfiability problem obtained by requiring that all the clauses in F have the same length $K_a = K$, is called the K**-satisfiability** (or K-SAT) problem.

As usual, an optimization problem is naturally associated to the decision version of satisfiability: Given a formula F, one is asked to find an assignment which violates the smallest number of clauses. This is called the **MAX-SAT** problem.

[1] It may happen that there does not exist any assignment satisfying F, so that one cannot use this indicator function to define a probability measure. However one can still characterize the local structure of $\mathbb{I}(\underline{x} \text{ satisfies } F)$ by the factor graph

Exercise 10.1 Consider the 2-SAT instance defined by the formula $F_1 = (x_1 \vee \overline{x}_2) \wedge (x_2 \vee \overline{x}_3) \wedge (\overline{x}_2 \vee x_4) \wedge (x_4 \vee \overline{x}_1) \wedge (\overline{x}_3 \vee \overline{x}_4) \wedge (\overline{x}_2 \vee x_3)$. Show that this formula is SAT and write a SAT-assignment.
[Hint: assign for instance $x_1 = 1$; the clause $x_4 \vee \overline{x}_1$ is then reduced to x_4, this is a **unit clause** which fixes $x_4 = 1$; the chain of 'unit clause propagation' either leads to a SAT assignment, or to a contradiction.]

Exercise 10.2 Consider the 2-SAT formula $F_2 = (x_1 \vee \overline{x}_2) \wedge (x_2 \vee \overline{x}_3) \wedge (\overline{x}_2 \vee x_4) \wedge (x_4 \vee \overline{x}_1) \wedge (\overline{x}_3 \vee \overline{x}_4) \wedge (\overline{x}_2 \vee \overline{x}_3)$. Show that this formula is UNSAT by using the same method as in the previous Exercise.

Exercise 10.3 Consider the 3-SAT formula $F_3 = (x_1 \vee x_2 \vee \overline{x}_3) \wedge (x_1 \vee x_3 \vee \overline{x}_4) \wedge (x_2 \vee x_3 \vee x_4) \wedge (\overline{x}_1 \vee x_2 \vee \overline{x}_4) \wedge (x_1 \vee \overline{x}_2 \vee x_4) \wedge (\overline{x}_1 \vee \overline{x}_2 \vee x_4) \wedge (\overline{x}_2 \vee \overline{x}_3 \vee \overline{x}_4) \wedge (x_2 \vee \overline{x}_3 \vee x_4) \wedge (\overline{x}_1 \vee x_3 \vee \overline{x}_4)$. Show that it is UNSAT.
[Hint: try to generalize the previous method by using a decision tree, cf. Sec. 10.2.2 below, or list the 16 possible assignments and cross out which one is eliminated by each clause.]

As we already mentioned, satisfiability was the first problem to be proved NP-complete. The restriction defined by requiring $K_a \leq 2$ for each clause a, is polynomial. However, if one relaxes this condition to $K_a \leq K$, with $K = 3$ or more, the resulting problem is NP-complete. For instance 3-SAT is NP-complete while 2-SAT is polynomial. It is intuitively clear that MAX-SAT is "at least as hard" as SAT: an instance is SAT if and only if the minimum number of violated clauses (that is the output of MAX-SAT) vanishes. It is less obvious that MAX-SAT can be "much harder" than SAT. For instance, MAX-2-SAT is NP-hard, while as said above, its decision counterpart is in P.

The study of applications is not the aim of this book, but one should keep in mind that satisfiability is related to a myriad of other problems, some of which have enormous practical relevance. It is a problem of direct importance to the fields of mathematical logic, computing theory and artificial intelligence. Applications range from integrated circuit design (modeling, placement, routing, testing,...) to computer architecture design (compiler optimization, scheduling and task partitioning,...) and to computer graphics, image processing etc...

10.2 Algorithms

10.2.1 A simple case: 2-SAT

The reader who worked out Exercises 10.1 and 10.2 has already a feeling that 2-SAT is an easy problem. The main tool for solving it is the so-called **unit clause propagation (UCP)** procedure. If we start from a 2-clause $C = z_1 \vee z_2$ and fix the literal z_1, two things may happen:

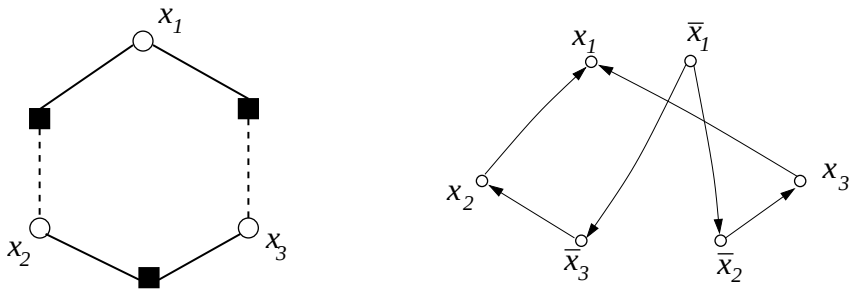

Fig. 10.2 Factor graph representation of the 2SAT formula $F = (x_1 \vee \overline{x}_2) \wedge (x_1 \vee \overline{x}_3) \wedge (x_2 \vee x_3)$ (left) and the corresponding directed graph $\mathcal{D}(F)$ (right).

- If we fix $z_1 = 1$ the clause is satisfied and disappears from the formula
- If we fix $z_1 = 0$ the clause is transformed into the unit clause z_2 which implies that $z_2 = 1$.

Given a 2-SAT formula, one can start from a variable x_i, $i \in \{1, \ldots, N\}$ and fix, for instance $x_i = 0$. Then apply the reduction rule described above to all the clauses in which x_i or $\overline{x}_i$ appears. Finally, fix recursively in the same way all the literals which appear in unit clauses. This procedure may halt for one of the following reasons: (i) the formula does not contain any unit clause; (ii) the formula contains the unit clause z_j together with its negation $\overline{z}_j$.

In the first case, a partial SAT assignment (i.e. an assignment of a subset of the variables such that no clause is violated) has been found. We will prove below that such a partial assignment can be extended to a complete SAT assignment if and only if the formula is SAT. One therefore repeats the procedure by fixing a not-yet-assigned variable x_j.

In the second case, the partial assignment cannot be extended to a SAT assignment. One proceeds by changing the initial choice and setting $x_i = 1$. Once again, if the procedure stops because of reason (i), then the formula can be effectively reduced and the already-fixed variables do not need to be reconsidered in the following. If on the other hand, also the choice $x_i = 1$ leads to a contradiction (i.e. the procedure stops because of (ii)), then the formula is UNSAT.

It is clear that the algorithm defined in this way is very efficient. Its complexity can be measured by the number of variable-fixing operations that it involves. Since each variable is considered at most twice, this number is at most $2N$.

For proving the correctness of this algorithm, we still have to show the following fact: if the formula is SAT and UCP stops because of reason (i), then the resulting partial assignment can be extended to a global SAT assignment (The implication in the reverse direction is obvious). The key point is that the residual formula is formed by a subset $\mathcal{R}$ of the variables (the ones which have not yet been fixed) together with *a subset of the original clauses* (those which involve uniquely variables in $\mathcal{R}$). If a SAT assignment exists, its restriction to $\mathcal{R}$ satisfies the residual formula and constitutes an extension of the partial assignment generated by UCP.

Exercise 10.4 Write a code for solving 2-SAT using the algorithm described above.

Exercise 10.5 A nice way to understand UCP, and why it is so effective for 2-SAT, consists in associating to the formula F a directed graph $\mathcal{D}(F)$ (not to be confused with the factor graph!) as follows. Associate a vertex to each of the $2N$ literals (for instance we have one vertex for x_1 and one vertex for $\overline{x}_1$). Whenever a clause like e.g. $\overline{x}_1 \vee x_2$ appears in the formula, we have two implications: if $x_1 = 1$ then $x_2 = 1$; if $x_2 = 0$ then $x_1 = 0$. Represent them graphically by drawing an directed edge from the vertex x_1 toward x_2, and an directed edge from $\overline{x}_2$ to $\overline{x}_1$.

Show that F is UNSAT if and only if there exists a variable index $i \in \{1, \dots, N\}$ such that: $\mathcal{D}(F)$ contains a directed path from x_i to $\overline{x}_i$, *and* a directed path from $\overline{x}_i$ to x_i. [Hint: Consider the UCP procedure described above and rephrase it in terms of the directed graph $\mathcal{D}(F)$. Show that it can be regarded as an algorithm for finding a pair of paths from x_i to $\overline{x}_i$ and vice-versa in $\mathcal{D}(F)$.]

Let us finally notice that the procedure described above does not give any clue about an efficient solution of MAX-2SAT, apart from determining whether the minimum number of violated clauses vanishes or not. As already mentioned MAX-2SAT is NP-hard.

10.2.2 A general complete algorithm

As soon as we allow an unbounded number of clauses of length 3 or larger, satisfiability becomes an NP-complete problem. Exercise 10.3 shows how the UCP strategy fails: fixing a variable in a 3-clause may leave a 2-clause. As a consequence, UCP may halt without contradictions and produce a residual formula containing clauses which were not present in the original formula. Therefore, it can be that the partial assignment produced by UCP cannot be extended to a global SAT assignment even if the original formula is SAT. Once a contradiction is found, it may be necessary to change any of the choices made so far in order to find a SAT assignment (in contrast to 2SAT where only the last choice had to be changed). The exploration of all such possibilities is most conveniently described through a decision tree. Each time a contradiction is found, the search algorithm backtracks to the last choice for which one possibility was not explored.

The most widely used **complete algorithms** (i.e. algorithms which are able to either find a satisfying assignment, or prove that there is no such assignment) rely on this idea. They are known under the name **DPLL**, from the initials of their inventors, Davis, Putnam, Logemann and Loveland. The basic recursive process is best explained on an example, as in Fig. 10.3. Its structure can be summarized in few lines, using the recursive procedure DPLL, which takes as input a CNF formula F, a partial assignment of the variables A, and the list of indices of unassigned variables V, and returns either 'UNSAT', or a SAT assignment. For solving a problem given by the CNF formula F, written in terms of the N variables $x_1, \cdots, x_N$, the initial call to this procedure should be DPLL(F, $\emptyset$, $\{1, \cdots, N\}$).

DPLL (formula F, partial assignment A, unassigned variables V)

1: if $V \neq \emptyset$:
2: Choose an index $i \in V$;
3: B=DPLL($F|\{x_i = 0\}$, $A \cup \{x_i = 0\}$, $V \setminus i$);
4: if B=UNSAT B=DPLL($F|\{x_i = 1\}$, $A \cup \{x_i = 1\}$, $V \setminus i$);
5: else **return** $A \cup \{x_i = 0\} \cup B$;
6: if B=UNSAT **return** B;
7: else **return** $A \cup \{x_i = 1\} \cup B$;
8: else:
9: if F has no clause **return** A;
10: else **return** UNSAT;

The notation $F|\{x_i = 0\}$ refers to the formula obtained from F by assigning x_i to 0: all clauses of F which contain the litteral $\overline{x}_i$ are eliminated, while clauses that contain x_i are shortened, namely $y \vee x_i$ is reduced to y. The reduced formula $F|\{x_i = 1\}$ is defined analogously.

As shown in Fig. 10.3 the algorithm can be represented as a walk in the decision tree. When it finds a contradiction, i.e. it reaches an 'UNSAT' leaf of the tree, it backtracks and searches a different branch.

In the above pseudo-code, we did not specify how to select the next variable to be fixed in step 2. Various versions of the DPLL algorithm differ in the order in which the variables are taken in consideration and the branching process is performed. Unit clause propagation can be rephrased in the present setting as the following rule: whenever the formula F contains clauses of length 1, x_i must be chosen among the variables appearing in such clauses. In such a case, no branching takes place. For instance, if the literal x_i appears in a unit clause, setting $x_i = 0$ would produce an empty clause and therefore a contradiction: one is forced to set $x_i = 1$.

Apart from the case of unit clauses, deciding on which variable the next branching will be done is an art, and can result in strongly varying performances. For instance, it is a good idea to branch on a variable which appears in many clauses, but other criteria, like the number of unit clauses that a branching will generate, can also be used. It is customary to characterize the performances of this class of algorithms by the number of branching nodes it generates. This does not correspond to the actual number of operations executed, which may depend on the heuristic. However, for many reasonable heuristics, the actual number of operations is within a polynomial factor (in the instance size) from the number of branchings and such a factor does not affect the leading exponential behavior.

Whenever the DPLL procedure does not return a SAT assignment, the formula is UNSAT: a representation of the explored search tree provides a proof. This is sometimes also called an UNSAT **certificate**. Notice that the length of an UNSAT certificate is (in general) larger than polynomial in the input size. This is at variance with a SAT certificate, which is provided, for instance, by a particular SAT assignment.

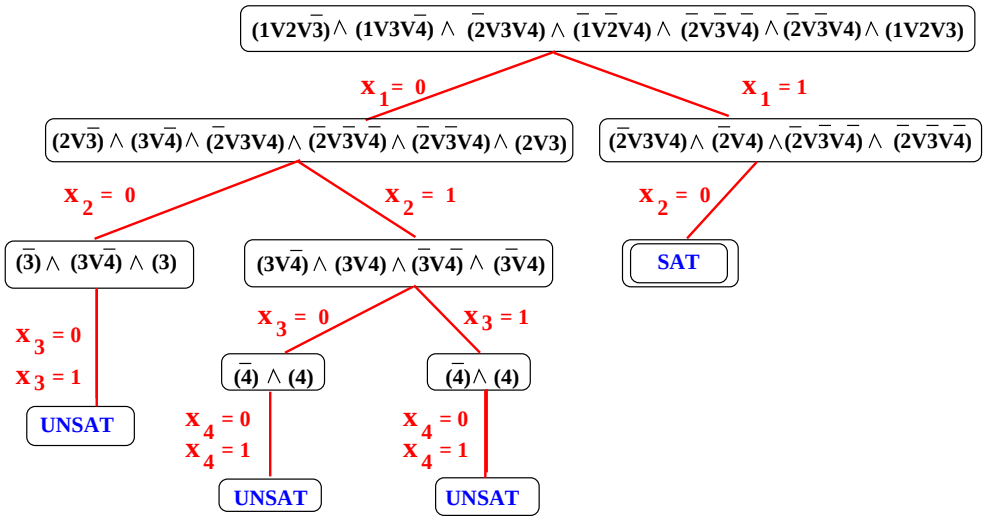

Fig. 10.3 A sketch of the DPLL algorithm, acting on the formula $(x_1 \vee x_2 \vee \overline{x}_3) \wedge (x_1 \vee x_3 \vee \overline{x}_4) \wedge (\overline{x}_2 \vee x_3 \vee x_4) \wedge (\overline{x}_1 \vee x_2 \vee x_4) \wedge (\overline{x}_2 \vee \overline{x}_3 \vee \overline{x}_4) \wedge (\overline{x}_2 \vee \overline{x}_3 \vee x_4) \wedge (x_1 \vee x_2 \vee x_3) \wedge (\overline{x}_1 \vee x_2 \vee \overline{x}_4)$. In order to get a more readable figure, the notation has been simplified: a clause like $(\overline{x}_1 \vee x_2 \vee x_4)$ is denoted here as $(\overline{1}\ 2\ 4)$. One fixes a first variable, here $x_1 = 0$. The problem is then reduced: clauses containing x_1 are eliminated, and clauses containing $\overline{x}_1$ are shortened by eliminating the literal $\overline{x}_1$. Then one proceeds by fixing a second variable, etc... At each step, if a unit clause is present, the next variable to be fixed is chosen among the those appearing in unit clauses. This corresponds to the unit clause propagation (UCP) rule. When the algorithm finds a contradiction (two unit clauses fixing a variable simultaneously to 0 and to 1), it backtracks to the last not-yet-explored branching node and explores another choice for the corresponding variable. In this case for instance, the algorithm first fixes $x_1 = 0$, then it fixes $x_2 = 0$, which implies through UCP that $x_3 = 0$ and $x_3 = 1$. This is a contradiction, and therefore the algorithm backtracks to the last choice, which was $x_2 = 0$, and tries instead the other choice: $x_2 = 1$, etc... Here, branching follows the order of appearence of variables in the formula.

Exercise 10.6 Resolution and DPLL.

(*i*) A powerful approach to proving that a formula is UNSAT relies on the idea of the **resolution proof**. Imagine that F contains two clauses: $x_j \vee A$, and $\overline{x}_j \vee B$, where A and B are subclauses. Show that these two clauses automatically imply the **resolvent on** x_j, that is the clause $A \vee B$.

(*ii*) A resolution proof is constructed by adding resolvent clauses to F. Show that, if this process produces an empty clause, then the original formula is necessarily UNSAT. An UNSAT certificate is simply given by the sequence of resolvents leading to the empty clause.

(*iii*) Although this may look different from DPLL, any DPLL tree is an example of resolution proof. To see this proceed as follows. Label each 'UNSAT' leave of the DPLL tree by the resolution of a pair of clauses of the original formula which are shown to be contradictory on this branch (e.g. the leftmost such leaf in Fig. 10.3 corresponds to the pair of initial clauses $x_1 \vee x_2 \vee \overline{x}_3$ and $x_1 \vee x_2 \vee x_3$, so that it can be labeled by the resolvent of these two clauses on x_3, namely $x_1 \vee x_2$). Show that each branching node of the DPLL tree can be labeled by a clause which is a resolvent of the two clauses labeling its children, and that this process, when carried on an UNSAT formula, produces a root (the top node of the tree) which is an empty clause.

10.2.3 Incomplete search

As we have seen above, proving that a formula is SAT is much easier than proving that it is UNSAT: one 'just' needs to exhibit an assignment that satisfies all the clauses. One can therefore relax the initial objective, and look for an algorithm that only tries to deal with the first task. This is often referred to as an **incomplete search** algorithm. Such an algorithm can either return a satisfying assignment or just say 'I do not know' whenever it is unable to find one (or to prove that the formula is UNSAT).

A basic algorithm for incomplete search, due to Schöning, is based on the simple random walk routine:

```
WALK (CNF formula F in N variables)
1:  for each variable i, set x_i = 0 or x_i = 1 with probability 1/2;
2:  repeat 3N times:
3:      if the current assignment satisfies F, return it and stop;
4:      else:
5:          choose an unsatisfied clause a uniformly at random;
6:          choose a variable index i uniformly at random in ∂a;
7:          flip the variable i (i.e.: x_i ← 1 − x_i);
8:  end
```

For this algorithm one can obtain a guarantee of performance:

Proposition 10.1 *Denote by $p(F)$ the probability that this routine, when executed on a formula F, returns a satisfying assignment. If F is SAT, then $p(F) \geq p_N$ where*

$$p_N = \frac{2}{3}\left(\frac{K}{2(K-1)}\right)^N . \tag{10.2}$$

One can therefore run the routine many times (with independent random numbers each time) in order to increase the probability of finding a solution. Suppose that the formula is SAT. If the routine is run $20/p_N$ times, the probability of not finding any solution is $(1-p_N)^{20/p_N} \leq e^{-20}$. While this is of course not a proof of unsatisfiability, it is very close to it. In general, the time required for this procedure to reduce the error probability below any fixed ε grows as

$$\tau_N \doteq \left(\frac{2(K-1)}{K}\right)^N . \tag{10.3}$$

This simple randomized algorithm achieves an exponential improvement over the naive exhaustive search which takes about 2^N operations.

Proof: Let us now prove the lower bound (10.2) on the probability of finding a satisfying assignment during a single run of the routine Walk. Since, by assumption, F is SAT, we can consider a particular SAT assignment, let us say $\underline{x}_*$. Let $\underline{x}_t$ be the

assignment produced by Walk(F) after t spin flips, and d_t be the Hamming distance between $\underline{x}_*$ and $\underline{x}_t$. Obviously, at time 0 we have

$$\mathbb{P}\{d_0 = d\} = \frac{1}{2^N}\binom{N}{d}. \tag{10.4}$$

Since $\underline{x}_*$ satisfies F, each clause is satisfied by at least one variable as assigned in $\underline{x}_*$. Mark *exactly* one such variable per clause. Each time Walk($\cdot$) chooses a violated clause, it flips a marked variable with probability $1/K$, reducing the Hamming distance by one. Of course, the Hamming distance can decrease also when another variable is flipped (if more than one variable in $\underline{x}_*$ satisfies this clause). In order to get a bound we introduce an auxiliary integer variable $\hat{d}_t$ which decreases by one each time a marked variable is selected, and increases by one (the maximum possible increase in Hamming distance due to a single flip) otherwise. If we choose the initial condition $\hat{d}_0 = d_0$, it follows from the previous observations that $d_t \le \hat{d}_t$ for any $t \ge 0$. We can therefore upper bound the probability that `Walk` finds a solution by the probability that $\hat{d}_t = 0$ for some $0 \le t \le 3N$. But the random process $\hat{d}_t = 0$ is simply a biased random walk on the half-line with initial condition (10.4): at each time step it moves to the left with probability $1/K$ and to the right with probability $1 - 1/K$. The probability of hitting the origin can then be estimated as in Eq. (10.2), as shown in the following exercise.

Exercise 10.7 Analysis of the biased random walk $\hat{d}_t$.

(*a*) Show that the probability for $\hat{d}_t$ to start at position d at $t = 0$ and be at the origin at time t is

$$\mathbb{P}\big\{\hat{d}_0 = d\,;\ \hat{d}_t = 0\big\} = \frac{1}{2^N}\binom{N}{d}\,\frac{1}{K^t}\binom{t}{\frac{t-d}{2}}\,(K-1)^{\frac{t-d}{2}} \tag{10.5}$$

for $t + d$ even, and vanishes otherwise.

(*b*) Use Stirling's formula to derive an approximation of this probability to the leading exponential order: $\mathbb{P}\big\{\hat{d}_0 = d\,;\ \hat{d}_t = 0\big\} \doteq \exp\{-N\Psi(\theta,\delta)\}$, where $\theta = t/N$ and $\delta = d/N$.

(*c*) Minimize $\Psi(\theta,\delta)$ with respect to $\theta \in [0,3]$ and $\delta \in [0,1]$, and show that the minimum value is $\Psi_* = \log[2(K-1)/K]$. Argue that $p_N \doteq \exp\{-N\Psi_*\}$ to the leading exponential order.

□

Notice that the above algorithm applies a very noisy strategy. While 'focusing' on unsatisfied clauses, it makes essentially random steps. The opposite philosophy would be that of making greedy steps. An example of 'greedy' step is the following: flip a variable which will lead to the largest positive increase in the number of satisfied clause.

There exist several refinements of the simple random walk algorithm. One of the greatest improvement consists in applying a mixed strategy: With probability p, pick

an unsatisfied clause, and flip a randomly chosen variable in this clause (as in Walk); With probability $1-p$, perform a 'greedy' step as defined above.

The pseudocode of this "Walksat" algorithm is given below, using the following notations: $E(\underline{x})$ is the number of clauses violated by assignment $\underline{x} = (x_1, \cdots, x_N)$ and $\underline{x}^{(i)}$ is the assignment obtained from $\underline{x}$ by flipping $x_i \to 1 - x_i$.

```
WalkSAT ( CNF formula F, number of flips f, mixing p )
 1 :  t = 0;
 2 :  Initialize x to a random assignment;
 3 :  While t < f do
 4 :     If x satisfies F, return x;
 5 :     Let r be uniformly random in [0, 1];
 6 :     If r < 1 − p then
 7 :        For each i ∈ V, let Δ_i = E(x^(i)) − E(x);
 8 :        Flip a variable x_i for which Δ_i is minimal;
 9 :     else
10:        Choose a violated clause a uniformly at random;
11:        Flip a uniformly random variable x_i, i ∈ ∂a;
12    End-While
13:  Return 'Not found';
```

This strategy works reasonably well if p is properly optimized. The greedy steps drive the assignment toward 'quasi-solutions', while the noise term allows to escape from local minima.

10.3 Random K-satisfiability ensembles

Satisfiability is NP-complete. One thus expects a complete algorithm to take exponential time in the worst case. However empirical studies have shown that many formulas are very easy to solve. A natural research direction is therefore to characterize ensembles of problems which are easy, separating them from those that are hard. Such ensembles can be defined by introducing a probability measure over the space of instances.

One of the most interesting family of ensembles is **random K-SAT**. An instance of random K-SAT contains only clauses of length K. The ensemble is further characterized by the number of variables N, and the number of clauses M, and denoted as $\mathsf{SAT}_N(K, M)$. A formula in $\mathsf{SAT}_N(K, M)$ is generated by selecting M clauses of size K uniformly at random among the $\binom{N}{K}2^K$ such clauses. Notice that the factor graph associated to a random K-SAT formula from the $\mathsf{SAT}_N(K, M)$ ensemble is in fact a random $\mathbb{G}_N(K, M)$ factor graph.

It turns out that a crucial parameter characterizing the random K-SAT ensemble is the **clause density** $\alpha \equiv M/N$. We shall define the 'thermodynamic' limit as $M \to \infty,\ N \to \infty$, with fixed density α. In this limit, several important properties of random formulas concentrate in probability around their typical values.

As in the case of random graphs, it is sometimes useful to consider slight variants

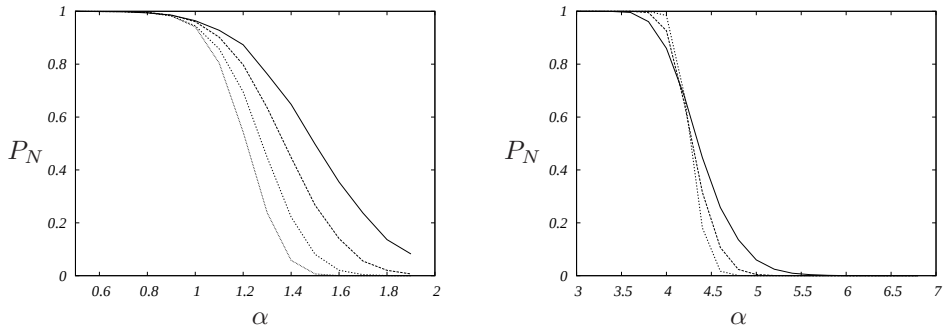

Fig. 10.4 Probability that a formula generated from the random K-SAT ensemble is satisfied, plotted versus the clause density α. Left: $K = 2$, right: $K = 3$. The curves have been generated using a DPLL algorithm. Each point is the result of averaging over 10^4 random formulas. The curves for $K = 2$ correspond to formulas of size $N = 50$, 100, 200, 400 (from right to left). In the case $K = 3$ the curves correspond to $N = 50$ (full line), $N = 100$ (dashed), $N = 200$ (dotted). The transition between satisfiable and unsatisfiable formulas becomes sharper as N increases.

of the above definition. One such variant is the $\mathsf{SAT}_N(K, \alpha)$ ensemble. A random instance from this ensemble is generated by including in the formula each of the $\binom{N}{K}2^K$ possible clauses independently with probability $\alpha N 2^{-K}/\binom{N}{K}$. Once again, the corresponding factor graph will be distributed according to the $\mathbb{G}_N(K, \alpha)$ ensemble introduced in Chapter 9. For many properties, differences between such variants vanish in the thermodynamic limit (this is analogous to the equivalence of different factor graph ensembles).

10.3.1 Numerical experiments

Using the DPLL algorithm, one can investigate the properties of typical instances of the random K-SAT ensemble $\mathsf{SAT}_N(K, M)$. Figure 10.4 shows the probability $P_N(K, \alpha)$ that a randomly generated formula is satisfiable, for $K = 2$ and $K = 3$. For fixed K and N, this is a decreasing function of α, which to 1 in the $\alpha \to 0$ limit and goes to 0 in the $\alpha \to \infty$ limit. One interesting feature in these simulations is the fact that the crossover from high to low probability becomes sharper and sharper when N increases. This numerical result points at the existence of a phase transition at a finite value $\alpha_s(K)$: for $\alpha < \alpha_s(K)$ ($\alpha > \alpha_s(K)$) a random K-SAT formula is SAT (respectively, UNSAT) with probability approaching 1 as $N \to \infty$.

The conjectured phase transition in random satisfiability problems with $K \geq 3$ has drawn considerable attention. One important reason comes from the study of the computational effort needed to solve the problem. Figure 10.5 shows the typical number of branching nodes in the DPLL tree required to solve a typical random 3-SAT formula. One may notice two important features: For a given value of the number of variables N, the computational effort has a peak in the region of clause density where a phase transition seems to occur (compare to Fig. 10.4). In this region it also increases rapidly with N. Looking carefully at the data one can distinguish qualitatively three

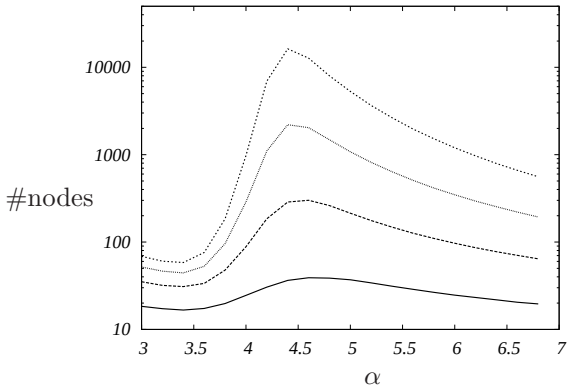

Fig. 10.5 Computational effort of our DPLL algorithm applied to random 3-SAT formulas. The logarithm of the number of branching nodes was averaged over 10^4 instances. From bottom to top: $N = 50$, 100, 150, 200.

different regions: at low α the solution is 'easily' found and the computer time grows polynomially; at intermediate α, in the phase transition region, the problem becomes typically very hard and the computer time grows exponentially. At larger α, in the region where a random formula is almost always UNSAT, the problem becomes easier, although the size of the DPLL tree still grows exponentially with N.

The hypothetical phase transition region is therefore the one where the hardest instances of random 3-SAT are located. This makes such a region particularly interesting, both from the point of view of computational complexity and from that of statistical physics.

10.4 Random 2-SAT

From the point of view of computational complexity, 2-SAT is polynomial while K-SAT is NP-complete for $K \geq 3$. It turns out that random 2-SAT is also much simpler to analyze than the other cases. One important reason is the existence of the polynomial decision algorithm described in Sec. 10.2.1 (see in particular Exercise 10.5). This can be analyzed in detail using the representation of a 2-SAT formula as a directed graph whose vertices are associated to literals. One can then use the mathematical theory of random directed graphs. In particular, the existence of a phase transition at critical clause density $\alpha_s(2) = 1$ can be established.

Theorem 10.2 *Let $P_N(K = 2, \alpha)$ the probability for a $\mathsf{SAT}_N(K = 2, M)$ random formula to be SAT. Then*

$$\lim_{N\to\infty} P_N(K = 2, \alpha) = \begin{cases} 1 & \text{if } \alpha < 1 \ , \\ 0 & \text{if } \alpha > 1 \ . \end{cases} \tag{10.6}$$

Proof: Here we shall prove that a random formula is SAT with high proability for $\alpha < 1$. It follows from theorem 10.5 below that it is with probability unsat for $\alpha > 1$.

We use the directed graph representation defined in Ex. 10.5. In this graph, define a bicycle of length s as a path $(u, w_1, w_2, \ldots, w_s, v)$, where the w_i's are literals on s distinct variables, and $u, v \in \{w_1, \ldots, w_s, \overline{w}_1, \ldots, \overline{w}_s\}$. As we saw in Ex. 10.5, if a formula F is UNSAT, its directed graph $\mathcal{D}(F)$ has a cycle containing the two literals x_i and $\overline{x}_i$ for some i. From such a cycle one easily builds a bicycle. The probability that a bicycle appears in $\mathcal{D}(F)$ is in turn upper bounded by the expected number of bicycles. Therefore:

$$\mathbb{P}(F \text{ is UNSAT}) \leq \mathbb{P}(\mathcal{D}(F) \text{ has a bicycle}) \leq \sum_{s=2}^{N} N^s 2^s (2s)^2 M^{s+1} \left(\frac{1}{4\binom{N}{2}} \right)^{s+1} . \tag{10.7}$$

The sum is over the size s of the bicycle; N^s is an upper bound to $\binom{N}{s}$, the number of ways one can choose the s variables; 2^s corresponds to the choice of literals, given the variables; $(2s)^2$ to the choice of u, v; M^{s+1} is an upper bound to $\binom{M}{s+1}$, the number of choices of the clauses involved in the bicycle; the last factor is the probability that each of the chosen clauses of the bicycle appears in the random formula. A direct summation of the series in 10.7 shows that, in the large N limit, the result is $O(1/N)$ whenever $\alpha < 1$. □

10.5 Phase transition in random $K(\geq 3)$-SAT

10.5.1 Satisfiability threshold conjecture

As noticed above, numerical studies suggest that random K-SAT undergoes a phase transition between a SAT phase and an UNSAT phase, for any $K \geq 2$. There is a widespread belief that this is indeed true, as formalized by the following conjecture:

Conjecture 10.3 *For any $K \geq 2$, there exists a threshold $\alpha_{\rm s}(K)$ such that:*

$$\lim_{N\to\infty} P_N(K, \alpha) = \begin{cases} 1 & \text{if } \alpha < \alpha_{\rm s}(K) \ , \\ 0 & \text{if } \alpha > \alpha_{\rm s}(K) \ . \end{cases} \tag{10.8}$$

As discussed in the previous section, this conjecture is proved in the case $K = 2$. The existence of a phase transition is still an open problem for larger K, although the following theorem gives some strong support:

Theorem 10.4. (Friedgut) *Let $P_N(K, \alpha)$ be the probability for a random formula from the $\mathsf{SAT}_N(K, M)$ ensemble to be satisfiable, and assume $K \geq 2$. Then there exists a sequence of $\alpha_{\rm s}^{(N)}(K)$ such that, for any $\varepsilon > 0$,*

$$\lim_{N\to\infty} P_N(K, \alpha_N) = \begin{cases} 1 & \text{if } \alpha_N < \alpha_{\rm s}^{(N)}(K) - \varepsilon \ , \\ 0 & \text{if } \alpha_N > \alpha_{\rm s}^{(N)}(K) + \varepsilon \ , \end{cases} \tag{10.9}$$

In other words, the crossover from SAT to UNSAT becomes sharper and sharper as N increases. For N large enough, it takes place in a window smaller than any fixed width ε. The 'only' missing piece to prove the satisfiability threshold conjecture (10.3) is the convergence of $\alpha_{\rm s}^{(N)}(K)$ to some value $\alpha_{\rm s}(K)$ as $N \to \infty$.

10.5.2 Upper bounds

Rigorous studies have allowed to establish bounds on the satisfiability threshold $\alpha_{\rm s}^{(N)}(K)$ in the large N limit. Upper bounds are obtained by using the first moment method. The general strategy is to introduce a function $U(F)$ acting on formulas such that:

$$U(F) = \begin{cases} 0 & \text{if } F \text{ is UNSAT,} \\ \geq 1 & \text{otherwise.} \end{cases} \tag{10.10}$$

Therefore, if F is a random K-SAT formula

$$\mathbb{P}\{F \text{ is SAT}\} \;\leq\; \mathbb{E}\, U(F) \;. \tag{10.11}$$

The inequality becomes an equality if $U(F) = \mathbb{I}(F \text{ is SAT})$. Of course, we do not know how to compute the expectation in this case. The idea is to find some function $U(F)$ which is simple enough that $\mathbb{E}\, U(F)$ can be computed, and with an expectation value that goes to zero as $N \to \infty$, for large enough α.

The simplest such choice is $U(F) = Z(F)$, the number of SAT assignments (this is the analogous of a "zero-temperature" partition function). The expectation $\mathbb{E}\, Z(F)$ is equal to the number of assignments, 2^N, times the probability that an assignment is SAT (which does not depend on the assignment). Consider for instance the all-zero assignment $x_i = 0$, $i = 1, \dots, N$. The probability that it is SAT is equal to the product of the probabilities that is satisfies each of the M clauses. The probability that the all-zero assignment satisfies a clause is $(1 - 2^{-K})$ because a K-clause excludes one among the 2^K assignments of variables which appear in it. Therefore

$$\mathbb{E}\, Z(F) = 2^N (1 - 2^{-K})^M = \exp\left[N \left(\log 2 + \alpha \log(1 - 2^{-K})\right)\right] \;. \tag{10.12}$$

This result shows that, for $\alpha > \alpha_{\rm UB,1}(K)$, where

$$\alpha_{\rm UB,1}(K) \equiv -\log 2 / \log(1 - 2^{-K}) \;, \tag{10.13}$$

$\mathbb{E}\, Z(F)$ is exponentially small at large N. Equation (10.11) implies that the probability of a formula being SAT also vanishes at large N for such an α:

Theorem 10.5 *If* $\alpha > \alpha_{\rm UB,1}(K)$*, then* $\lim_{N\to\infty} \mathbb{P}\{F \text{ is SAT}\} = 0$*, whence* $\alpha_{\rm s}^{(N)}(K) \leq \alpha_{\rm UB,1}(K)$.

One should not expect this bound to be tight. The reason is that, in the SAT phase, $Z(F)$ takes exponentially large values, and its fluctuations tend to be exponential in the number of variables.

Example 10.6 As a simple illustration consider a toy example: the random 1-SAT ensemble $\mathsf{SAT}_N(1,\alpha)$. A formula is generated by including each of the $2N$ literals as a clause independently with probability $\alpha/2$ (we assume $\alpha \leq 2$). In order for the formula to be SAT, for each of the N variables, at most 1 of the 2 corresponding literals must be included. We have therefore

$$P_N(K=1,\alpha) = (1-\alpha^2/4)^N \,. \tag{10.14}$$

In other words, the probability for a random formula to be SAT goes exponentially fast to 0 for any $\alpha > 0$: $\alpha_{\rm s}(K=1) = 0$. On the other hand the upper bound deduced from $\mathbb{E}Z(F)$ is $\alpha_{\rm UB,1}(K) = 1$. This is due to large fluctuations in the number of SAT assignments Z, as we will see in the next exercise.

Exercise 10.8 Consider the distribution of $Z(F)$ in the random 1-SAT ensemble.

(*a*) Show that:

$$\mathbb{P}\{Z(F) = 2^n\} = \binom{N}{n} \left(1-\frac{\alpha}{2}\right)^{2n} \left[\alpha\left(1-\frac{\alpha}{4}\right)\right]^{N-n} , \tag{10.15}$$

for any $n \geq 0$.
[Hint: If F is SAT, then $Z(F) = 2^n$, where n is the number of variables which do not appear in any clause].

(*b*) From this expression, deduce the large deviation principle :

$$\mathbb{P}\left\{Z(F) = 2^{N\nu}\right\} \doteq \exp\{-N\, I_\alpha(\nu)\} \tag{10.16}$$

where:

$$I_\alpha(\nu) \equiv -\mathcal{H}(\nu) - 2\nu\log(1-\alpha/2) - (1-\nu)\log(\alpha(1-\alpha/4)) \,. \tag{10.17}$$

What is the most probable value of ν?

(*c*) Show that:

$$\mathbb{E}\, Z(F) \doteq \exp\left\{N \max_\nu [-I_\alpha(\nu) + \nu\log 2]\right\} \,. \tag{10.18}$$

What is the value of ν where the maximum is achieved, ν^*? Show that $I_\alpha(\nu^*) > 0$: the probability of having $Z(F) \doteq 2^{N\nu^*}$ is exponentially small, therefore $\mathbb{E}\, Z(F)$ is dominated by rare events.

Exercise 10.9 Repeat the derivation of Theorem 10.5 for the $\mathsf{SAT}_N(K,\alpha)$ ensemble (i.e. compute $\mathbb{E}\,Z(F)$ for this ensemble and find for which values of α this expectation is exponentially small). Show that the upper bound obtained in this case is $\alpha = 2^K \log 2$. This is worse than the previous upper bound $\alpha_{\rm UB,1}(K)$, although one expects the threshold to be the same. Why?
[Hint: The number of clauses M in a $\mathsf{SAT}_N(K,\alpha)$ formula has binomial distribution with parameters N, and α. What values of M provide the dominant contribution to $\mathbb{E}\,Z(F)$?]

In order to improve upon Theorem 10.5 using the first moment method, one needs a better (but still simple) choice of the function $U(F)$. A possible strategy consists in defining some small subclass of 'special' SAT assignments, such that if a SAT assignment exists, then a special SAT assignment exists too. If the subclass is small enough, one can hope to reduce the fluctuations in $U(F)$ and sharpen the bound.

One choice of such a subclass consists in 'locally maximal' SAT assignments. Given a formula F, an assignment $\underline{x}$ for this formula is said to be a locally maximal SAT assignment if and only if: (1) It is a SAT assignment, (2) for any i such that $x_i = 0$, the assignment obtained by flipping it to $x_i = 1$ is UNSAT. Define $U(F)$ as the number of locally maximal SAT assignments and apply the first moment method to this function. This gives:

Theorem 10.7 *For any $K \geq 2$, let $\alpha_{\rm UB,2}(K)$ be the unique positive solution of the equation:*

$$\alpha \log(1-2^{-K}) + \log\left[2 - \exp\left(-\frac{K\alpha}{2^K-1}\right)\right] = 0\,. \tag{10.19}$$

Then $\alpha_{\rm s}^{(N)}(K) \leq \alpha_{\rm UB,2}(K)$ for large enough N.

The proof is left as the following exercise:

Exercise 10.10 Consider an assignment $\underline{x}$ where exactly L variables are set to 0, the remaining $N-L$ ones being set to 1. Without loss of generality, assume $x_1, \dots, x_L$ to be the variables set to zero.

(*a*) Let p be the probability that a clause constrains the variable x_1, *given that* the clause is satisfied by the assignment $\underline{x}$ (By a clause constraining x_1, we mean that the clause becomes unsatisfied if x_1 is flipped from 0 to 1). Show that $p = \binom{N-1}{K-1}[(2^K-1)\binom{N}{K}]^{-1}$.

(*b*) Show that the probability that variable x_1 is constrained by at least one of the M clauses, given that all these clauses are satisfied by the assignment $\underline{x}$, is equal to $q = 1-(1-p)^M$

(*c*) Let $\mathcal{C}_i$ be the event that x_i is constrained by at least one of the M clauses. If $\mathcal{C}_1$, ..., $\mathcal{C}_L$ were independent events, under the condition that $\underline{x}$ satisfies F, the probability that $x_1, \dots x_L$ are constrained would be equal q^L. Of course $\mathcal{C}_1$, ..., $\mathcal{C}_L$ are not independent. Find an heuristic argument to show that they are anti-correlated and their joint probability is *at most* q^L (consider for instance the case $L=2$).

(*d*) Assume the claim at previous point to be true. Show that $\mathbb{E}[U(F)] \leq (1-2^{-K})^M \sum_{L=0}^{N}\binom{N}{L} q^L = (1-2^{-K})^M [1+q]^N$ and finish the proof by working out the large N asymptotics of this formula (with $\alpha = M/N$ fixed).

In Table 10.1 we report the numerical values of the upper bounds $\alpha_{\rm UB,1}(K)$ and $\alpha_{\rm UB,2}(K)$ for a few values of K. These results can be slightly improved upon by pursuing the same strategy. For instance, one may strengthen the condition of maximality to flipping 2 or more variables. However the quantitative improvement in the bound is rather small.

10.5.3 Lower bounds

Two main strategies have been used to derive lower bounds of $\alpha_{\rm c}^{(N)}(K)$ in the large N limit. In both cases one takes advantage of Theorem 10.4: In order to show that $\alpha_{\rm c}^{(N)}(K) \geq \alpha^*$, it is sufficient to prove that a random $\mathsf{SAT}_N(K,M)$ formula, with $M=\alpha N$, is SAT with non vanishing probability in the $N\to\infty$ limit.

The first approach consists in analyzing explicit heuristic algorithms for finding SAT assignments. The idea is to prove that a particular algorithm finds a SAT assignment with positive probability as $N\to\infty$ when α is smaller than some value.

One of the simplest such bounds is obtained by considering unit clause propagation. Whenever there exist a unit clause, assign the variables appearing in one of them in such a way to satisfy it, and proceed recursively. Otherwise, chose a variable uniformly at random among those which are not yet fixed and assign it to 0 or 1 with probability 1/2. The algorithm halts if it finds a contradiction (i.e. a couple of opposite unit clauses) or if all the variables have been assigned. In the latter case, the assignment produced by the algorithm satisfies the formula.

This algorithm is then applied to a random K-SAT formula with clause density α. It can be shown that a SAT assignment is found with positive probability for α small enough: this gives the lower bound $\alpha_{\rm c}^{(N)}(K) \geq \frac{1}{2}\left(\frac{K-1}{K-2}\right)^{K-2}\frac{2^K}{K}$ in the $N\to\infty$ limit.

In the Exercise below we give the main steps of the reasoning for the case $K = 3$, referring to the literature for more detailed proofs.

Exercise 10.11 After T iterations, the formula will contain 3-clauses, as well as 2-clauses and 1-clauses. Denote by $\mathcal{C}_s(T)$ the set of s-clauses, $s = 1, 2, 3$, and by $C_s(T) \equiv |\mathcal{C}_s(T)|$ its size. Let $\mathcal{V}(T)$ be the set of variables which have not yet been fixed, and $\mathcal{L}(T)$ the set of literals on the variables of $\mathcal{V}(T)$ (obviously we have $|\mathcal{L}(T)| = 2|\mathcal{V}(T)| = 2(N-T)$). Finally, if a contradiction is encountered after $T_{\rm halt}$ steps, we adopt the convention that the formula remains unchanged for all $T \in \{T_{\rm halt}, \dots, N\}$.

(*a*) Show that, for any $T \in \{1, \dots, N\}$, each clause in $\mathcal{C}_s(T)$ is uniformly distributed among the s-clauses over the literals in $\mathcal{L}(T)$.

(*b*) Show that the expected change in the number of 3- and 2-clauses is given by $\mathbb{E}\left[C_3(T+1) - C_3(T)\right] = -\frac{3C_3(T)}{N-T}$ and $\mathbb{E}\left[C_2(T+1) - C_2(T)\right] = \frac{3C_3(T)}{2(N-T)} - \frac{2C_2(T)}{N-T}$.

(*c*) Show that, conditional on $C_1(T)$, $C_2(T)$, and $C_3(T)$, the change in the number of 1-clauses is distributed as follows: $C_1(T+1) - C_1(T) \stackrel{\rm d}{=} -\mathbb{I}(C_1(T) > 0) + B\left(C_2(T), \frac{1}{N-T}\right)$. (We recall that $B(n,p)$ denotes a binomial random variable of parameters n, and p (cf. App. A)).

(*d*) It can be shown that, as $N \to \infty$ at fixed $t = T/N$, the variables $C_s(T)/N$ for $s \in \{2, 3\}$ concentrate around their expectation values, and these converge to smooth functions $c_s(t)$. Argue that these functions must solve the ordinary differential equations: $\frac{dc_3}{dt} = -\frac{3}{1-t}c_3(t)$; $\frac{dc_2}{dt} = \frac{3}{2(1-t)}c_3(t) - \frac{2}{1-t}c_2(t)$. Check that the solutions of these equations are: $c_3(t) = \alpha(1-t)^3$, $c_2(t) = (3\alpha/2)t(1-t)^2$.

(*e*) Show that the number of unit clauses is a Markov process described by $C_1(0) = 0$, $C_1(T+1) - C_1(T) \stackrel{\rm d}{=} -\mathbb{I}(C_1(T) > 0) + \eta(T)$, where $\eta(T)$ is a Poisson distributed random variable with mean $c_2(t)/(1-t)$, where $t = T/N$. Given C_1 and a time T, show that the probability that there is no contradiction generated by the unit clause algorithm up to time T is $\prod_{\tau=1}^{T}\left(1 - 1/(2(N-\tau))\right)^{[C_1(\tau)-1]\mathbb{I}(C_1(\tau)\geq 1)}$.

(*f*) Let $\rho(T)$ be the probability that there is no contradiction up to time T. Consider $T = N(1-\epsilon)$; show that $\rho(N(1-\epsilon)) \geq (1 - 1/(2N\epsilon))^{AN+B}\ \mathbb{P}(\sum_{\tau=1}^{N(1-\epsilon)} C_1(\tau) \leq AN + B)$. Assume that α is such that, $\forall t \in [0, 1-\epsilon]:\ c_2(t)/(1-t) < 1$. Show that there exists A, B such that $\lim_{N\to\infty} \mathbb{P}(\sum_{\tau=1}^{N(1-\epsilon)} C_1(\tau) \leq AN + B)$ is finite. Deduce that in the large N limit, there is a finite probability that, at time $N(1-\epsilon)$, the unit clause algorithm has not produced any contradiction so far, and $C_1(N(1-\epsilon)) = 0$.

(*g*) Conditionnaly to the fact that the algorithm has not produced any contradiction and $C_1(N(1-\epsilon)) = 0$, consider the residual formula at time $T = N(1-\epsilon)$. Transform each 3-clause into a 2-clause by removing from it a uniformly random variable. Show that one obtains, for ϵ small enough, a random 2-SAT problem with a small clause density $\leq 3\epsilon^2/2$, so that this is a satisfiable instance.

(*h*) Deduce that, for $\alpha < 8/3$, the unit clause propagation algorithm finds a solution with a finite probability

More refined heuristics have been analyzed using this method and lead to better

lower bounds on $\alpha_{\rm c}^{(N)}(K)$. We shall not elaborate on this approach here, but rather present a second strategy, based on a structural analysis of the problem. The idea is to use the second moment method. More precisely, we consider a function $U(F)$ of the SAT formula F, such that $U(F) = 0$ whenever F is UNSAT and $U(F) > 0$ otherwise. We then make use of the following inequality:

$$\mathbb{P}\{F \text{ is SAT}\} \ = \mathbb{P}\{U(F) > 0\} \ \geq \ \frac{[\mathbb{E}\, U(F)]^2}{\mathbb{E}[U(F)^2]} \ . \tag{10.20}$$

The present strategy is more delicate to implement than the first moment method, used in Sec. 10.5.2 to derive upper bounds on $\alpha_{\rm c}^{(N)}(K)$. For instance, the simple choice $U(F) = Z(F)$ does not give any result: It turns out that the ratio $[\mathbb{E}\, Z(F)]^2/\mathbb{E}[Z(F)^2]$ is exponentially small in N for any non vanishing value of α, so that the inequality (10.20) is useless. Again one needs to find a function $U(F)$ whose fluctuations are smaller than for the number $Z(F)$ of SAT assignments. More precisely, one needs the ratio $[\mathbb{E}\, U(F)]^2/\mathbb{E}[U(F)^2]$ to be non vanishing in the $N \to \infty$ limit.

A successful idea uses a weighted sum of SAT assignments:

$$U(F) = \sum_{\underline{x}} \prod_{a=1}^{M} W(\underline{x}, a) \ . \tag{10.21}$$

Here the sum is over all the 2^N assignments, and $W(\underline{x}, a)$ is a weight associated with clause a. This weight must be such that $W(\underline{x}, a) = 0$ when the assignment $\underline{x}$ does not satisfy clause a, and $W(\underline{x}, a) > 0$ otherwise. Let us choose a weight which depends on the number $r(\underline{x}, a)$ of variables which satisfy clause a in the assignment $\underline{x}$:

$$W(\underline{x}, a) = \begin{cases} \varphi(r(\underline{x}, a)) & \text{if } r(\underline{x}, a) \geq 1, \\ 0 & \text{otherwise.} \end{cases} \tag{10.22}$$

It is then easy to compute the first two moments of $U(F)$:

$$\mathbb{E}\, U(F) = 2^N \left[2^{-K} \sum_{r=1}^{K} \binom{K}{r} \varphi(r) \right]^M \ , \tag{10.23}$$

$$\mathbb{E}\left[U(F)^2\right] = 2^N \sum_{L=0}^{N} \binom{N}{L} \left[g_\varphi(N, L) \right]^M \ . \tag{10.24}$$

Here $g_\varphi(N, L)$ is the expectation value of the product $W(\underline{x}, a) W(\underline{y}, a)$ when a clause a is chosen uniformly at random, given that $\underline{x}$ and $\underline{y}$ are two assignments of N variables which agree on *exactly* L of them.

In order to compute $g_\varphi(N, L)$, it is convenient to introduce two binary vectors $\vec{u}, \vec{v} \in \{0, 1\}^K$. They encode the following information: Consider a clause a, fix $u_s = 1$ if in the assignment $\underline{x}$ the s-th variable of clause a satisfies the clause, and $u_s = 0$ otherwise. The components of $\vec{v}$ are defined similarly but with the assignment $\underline{y}$.

Furthermore, we denote by $d(\vec{u},\vec{v})$ the Hamming distance between these vectors, and by $w(\vec{u})$, $w(\vec{v})$ their Hamming weights (number of non zero components). Then

$$g_\varphi(N,L) = 2^{-K} {\sum_{\vec{u},\vec{v}}}' \, \varphi\left(w(\vec{u})\right) \varphi\left(w(\vec{v})\right) \left(\frac{L}{N}\right)^{d(\vec{u},\vec{v})} \left(1-\frac{L}{N}\right)^{K-d(\vec{u},\vec{v})} . \qquad (10.25)$$

Here the sum $\sum'$ runs over K-component vectors $\vec{u}$, $\vec{v}$ with at least one non zero component. A particularly simple choice is $\varphi(r) = \lambda^r$. Denoting $z = L/N$, one finds:

$$g_w(N,L) = 2^{-K} \left(\left[(\lambda^2+1)z + 2\lambda(1-z)\right]^K - 2\left[z+\lambda(1-z)\right]^K + z^k \right) . \qquad (10.26)$$

The first two moments can be evaluated from Eqs. (10.23), (10.24):

$$\mathbb{E}\, U(F) \doteq \exp\{N h_1(\lambda,\alpha)\} , \quad \mathbb{E}\left[U(F)^2\right] \doteq \exp\{N \max_z h_2(\lambda,\alpha,z)\} , \qquad (10.27)$$

where the maximum is taken over $z \in [0,1]$ and

$$h_1(\lambda,\alpha) \equiv \log 2 - \alpha K \log 2 + \alpha \log\left[(1+\lambda)^K - 1\right] , \qquad (10.28)$$

$$h_2(\lambda,\alpha,z) \equiv \log 2 - z\log z - (1-z)\log(1-z) - \alpha K \log 2 + \qquad (10.29)$$
$$+\alpha \log\left(\left[(\lambda^2+1)z + 2\lambda(1-z)\right]^K - 2\left[z+\lambda(1-z)\right]^K + z^k \right) .$$

Evaluating the above expression for $z = 1/2$ one finds $h_2(\lambda,\alpha,1/2) = 2h_1(\lambda,\alpha)$. The interpretation is as follows. Setting $z = 1/2$ amounts to assuming that the second moment of $U(F)$ is dominated by completely uncorrelated assignments (two uniformly random assignments agree on about half of the variables). This results in the factorization of the expectation $\mathbb{E}\left[U(F)^2\right] \approx \left[\mathbb{E}\, U(F)\right]^2$.

Two cases are possible: either the maximum of $h_2(\lambda,\alpha,z)$ over $z \in [0,1]$ is achieved only at $z = 1/2$ or not.

(*i*) In the latter case $\max_z h_2(\lambda,\alpha,z) > 2h_1(\lambda,\alpha)$ strictly, and therefore the ratio $[\mathbb{E}\, U(F)]^2/\mathbb{E}[U(F)^2]$ is exponentially small in N, the second moment inequality (10.20) is useless.

(*ii*) If on the other hand the maximum of $h_2(\lambda,\alpha,z)$ is achieved only at $z = 1/2$, then the ratio $[\mathbb{E}\, U(F)]^2/\mathbb{E}[U(F)^2]$ is 1 to the leading exponential order. It is not difficult to work out the precise asymptotic behavior (i.e. to compute the prefactor of the exponential). One finds that $[\mathbb{E}\, U(F)]^2/\mathbb{E}[U(F)^2]$ remains finite when $N \to \infty$. As a consequence $\alpha \le \alpha_{\rm c}^{(N)}(K)$ for N large enough.

A necessary condition for the second case to occur is that $z = 1/2$ is a local maximum of $h_2(\lambda,\alpha,z)$. This implies that λ must be the (unique) strictly positive root of:

$$(1+\lambda)^{K-1} = \frac{1}{1-\lambda} . \qquad (10.30)$$

We have thus proved that:

K	3	4	5	6	7	8	9	10
$\alpha_{\rm LB}(K)$	2.548	7.314	17.62	39.03	82.63	170.6	347.4	701.5
$\alpha_{\rm UB,1}(K)$	5.191	10.74	21.83	44.01	88.38	177.1	354.5	709.4
$\alpha_{\rm UB,2}(K)$	4.666	10.22	21.32	43.51	87.87	176.6	354.0	708.9

Table 10.1 Satisfiability thresholds for random K-SAT. We report the lower bound from Theorem 10.8 and the upper bounds from Eqs. (10.13) and (10.19).

Theorem 10.8 *Let λ be the positive root of Eq. (10.30), and the function $h_2(\,\cdot\,)$ be defined as in Eq. (10.29). Assume that $h_2(\lambda,\alpha,z)$ achieves its maximum, as a function of $z\in[0,1]$ only at $z=1/2$. Then a random $\mathsf{SAT}_N(K,\alpha)$ is SAT with probability approaching one as $N\to\infty$.*

Let $\alpha_{\rm LB}(K)$ be the largest value of α such that the hypotheses of this Theorem are satisfied. The Theorem implies an explicit lower bound on the satisfiability threshold: $\alpha_{\rm s}^{(N)}(K)\geq\alpha_{\rm LB}(K)$ in the $N\to\infty$ limit. Table 10.1 summarizes some of the values of the upper and lower bounds found in this Section for a few values of K. In the large K limit the following asymptotic behaviors can be shown to hold:

$$\alpha_{\rm LB}(K) = 2^K\log 2 - 2(K+1)\log 2 - 1 + o(1)\,, \tag{10.31}$$

$$\alpha_{\rm UB,1}(K) = 2^K\log 2 - \frac{1}{2}\log 2 + o(1)\,. \tag{10.32}$$

In other words, the simple methods exposed in this Chapter allow to determine the satisfiability threshold with a relative error behaving as 2^{-K} in the large K limit. More sophisticated tools, to be discussed in the next Chapters, are necessary for obtaining sharp results at finite K.

Exercise 10.12 [Research problem] Show that the choice of weight $\varphi(r)=\lambda^r$ is optimal: all other choices for $\varphi(r)$ give a worse lower bound. What strategy could be followed to improve the bound further?

Notes

The review paper (Gu, Purdom, Franco and Wah, 1996) is a rather comprehensive source of information on the algorithmic aspects of satisfiability. The reader interested in applications will also find there a detailed and referenced list.

Davis and Putnam first studied an algorithm for satisfiability in (Davis and Putnam, 1960). This was based on a systematic application of the resolution rule. The backtracking algorithm discussed in the main text was introduced in (Davis, Logemann and Loveland, 1962).

Other ensembles of random CNF formulas have been studied, but it turns out it is not so easy to find hard formulas. For instance take N variables, and generate M clauses independently according to the following rule. In a clause a, each of the variables appears as x_i or $\overline{x}_i$ with the same probability $p\leq 1/2$, and does not appear

with probability $1 - 2p$. The reader is invited to study this ensemble; an introduction and guide to the corresponding literature can be found in (Franco, 2000). Another useful ensemble is the "$2 + p$" SAT problem which interpolates between $K = 2$ and $K = 3$ by picking pM 3-clauses and $(1 - p)M$ 2-clauses, see (Monasson and Zecchina, 1998; Monasson, Zecchina, Kirkpatrick, Selman and Troyansky, 1999)

The polynomial nature of 2-SAT is discussed in (Cook, 1971). MAX-2SAT was shown to be NP-complete in (Garey, Johnson and Stockmeyer, 1976).

Schöning's algorithm was introduced in (Schöning, 1999) and further discussed in (Schöning, 2002). More general random walk strategies for SAT are treated in (Papadimitriou, 1991; Selman and Kautz, 1993; Selman, Kautz and Cohen, 1994).

The threshold $\alpha_s = 1$ for random 2-SAT was proved in (Chvátal and Reed, 1992), (Goerdt, 1996) and (de la Vega, 1992), see also (de la Vega, 2001). The scaling behavior near to the threshold has been analyzed through graph theoretical methods in (Bollobas, Borgs, Chayes, Kim and Wilson, 2001).

The numerical identification of the phase transition in random 3-SAT, and the observation that difficult formulas are found near the phase transition, were done in (Kirkpatrick and Selman, 1994; Selman and Kirkpatrick, 1996). See also (Selman, Mitchell and Levesque, 1996).

Friedgut's theorem is proved in (Friedgut, 1999).

Upper bounds on the threshold are discussed in (Dubois and Boufkhad, 1997; Kirousis, Kranakis, Krizanc and Stamatiou, 1998). Lower bounds for the threshold in random K-SAT based on the analysis of some algorithms were pioneered by Chao and Franco. The paper (Chao and Franco, 1986) corresponds to Exercise 10.11, and a generalization can be found in (Chao and Franco, 1990). A review of this type of methods is provided by (Achlioptas, 2001). Backtracking algorithms were first analyzed using an heuristic approach in (Cocco and Monasson, 2001*b*; Cocco and Monasson, 2001*a*). (Cocco, Monasson, Montanari and Semerjian, 2006) gives a survey of the analysis of algorithms based on statistical physics methods.

The idea of deriving a lower bound with the weighted second moment method was introduced in (Achlioptas and Moore, 2007). The lower bound which we discuss here is derived in (Achlioptas and Peres, 2004); this paper also solves the first question of Exercise 10.12. A simple introduction to the second moment method in various constraint satisfaction problems is (Achlioptas, Naor and Peres, 2005), see also (Gomes and Selman, 2005).

11
Low-Density Parity-Check Codes

Low-density parity-check (LDPC) error correcting codes were introduced in 1963 by Robert Gallager in his Ph.D. thesis. The basic motivation came from the observation that random linear codes, cf. Sec. 6.6, had excellent theoretical performances (in terms of the number of channel errors they could correct) but were unpractical. In particular, no efficient algorithm was known for decoding. In retrospect, this is not surprising, since it was later shown that decoding for linear codes is an NP-hard problem.

The idea was then to restrict the random linear code ensemble, introducing some structure that can be exploited for more efficient decoding. Of course, the risk is that such a restriction of the ensemble might spoil its performances. Gallager's proposal was simple and successful (but ahead of times): LDPC codes are among the most efficient codes around.

In this chapter we introduce one of the most important families of LDPC ensembles and derive some of its basic properties. As for any code, one can take two quite different points of view. The first is to study the code performances with respect to an appropriate metric, under *optimal* decoding, in which no constraint is imposed on the computational complexity of decoding procedure. For instance decoding through a scan of the whole, exponentially large, codebook is allowed. The second approach consists in analyzing the code performance under some specific, efficient, decoding algorithm. Depending on the specific application, one can be interested in algorithms of polynomial complexity, or even require the complexity to be linear in the block-length.

Here we will focus on performances under optimal decoding. We will derive rigorous bounds, showing that appropriately chosen LDPC ensembles allow to communicate reliably at rates close to Shannon's capacity. However, the main interest of LDPC codes is that they can be decoded efficiently, and we will discuss a simple example of decoding algorithm with linear time complexity. A more extensive study of LDPC codes under practical decoding algorithms is deferred to Ch. 15.

After defining LDPC codes and LDPC code ensembles in Section 11.1, we discuss some geometric properties of their codebooks in Section 11.2. In Section 11.3 we use these properties to derive a lower bound on the threshold for reliable communication. An upper bound follows from information-theoretic considerations. Section 11.4 discusses a simple decoding algorithm, which is shown to correct a finite fraction of errors.

11.1 Definitions

11.1.1 Linear algebra with binary variables

Remember that a code is characterized by its codebook $\mathfrak{C}$, which is a subset of $\{0,1\}^N$. LDPC codes are **linear codes**, which means that the codebook is a linear subspace of $\{0,1\}^N$. In practice such a subspace can be specified through an $M \times N$ matrix $\mathbb{H}$, with binary entries $\mathbb{H}_{ij} \in \{0,1\}$, and $M < N$. The codebook is defined as the kernel of $\mathbb{H}$:

$$\mathfrak{C} = \{\, \underline{x} \in \{0,1\}^N \;:\; \mathbb{H}\underline{x} = \underline{0} \,\}\,. \tag{11.1}$$

Here and in all this chapter, the multiplications and sums involved in $\mathbb{H}\underline{x}$ are understood as being computed modulo 2. The matrix $\mathbb{H}$ is called the **parity check matrix** of the code. The size of the codebook is $|\mathfrak{C}| = 2^{N-\mathrm{rank}(\mathbb{H})}$, where $\mathrm{rank}(\mathbb{H})$ denotes the rank of the matrix $\mathbb{H}$ (the number of linearly independent rows). As $\mathrm{rank}(\mathbb{H}) \leq M$, we have $|\mathfrak{C}| \geq 2^{N-M}$. With a slight modification with respect to the notation of Chapter 1, we let $L \equiv N - M$. The rate R of the code verifies therefore $R \geq L/N$, equality being obtained when all the rows of $\mathbb{H}$ are linearly independent.

Given such a code, encoding can always be implemented as a linear operation. There exists a $N \times L$ binary matrix $\mathbb{G}$, called the generator matrix, such that the codebook is the image of $\mathbb{G}$: $\mathfrak{C} = \{\underline{x} = \mathbb{G}\underline{z}\,,$ where $\underline{z} \in \{0,1\}^L\}$. Encoding is therefore realized as the mapping $\underline{z} \mapsto \underline{x} = \mathbb{G}\underline{z}$. (Notice that the product $\mathbb{H}\,\mathbb{G}$ is a $M \times L$ 'null' matrix with all entries equal to zero).

11.1.2 Factor graph

In Sect. 9.1.2 we described the factor graph associated with one particular linear code (the Hamming code of (9.8)). The recipe to build the factor graph, knowing $\mathbb{H}$, is as follows. Let us denote by $i^a_1, \ldots, i^a_{k(a)} \in \{1, \ldots, N\}$ the column indices such that $\mathbb{H}$ has a matrix element equal to 1 at row a and column i^a_j. Then the a-th coordinate of the vector $\mathbb{H}\underline{x}$ is equal to $x_{i^a_1} \oplus \cdots \oplus x_{i^a_{k(a)}}$. Let $\mu_{0,\mathbb{H}}(\underline{x})$ be the uniform distribution over all codewords of the code $\mathbb{H}$ (hereafter we shall often identify a code with its parity check matrix). It is given by:

$$\mu_{0,\mathbb{H}}(\underline{x}) = \frac{1}{Z} \prod_{a=1}^{M} \mathbb{I}(x_{i^a_1} \oplus \cdots \oplus x_{i^a_k} = 0)\,. \tag{11.2}$$

Therefore, the factor graph associated with $\mu_{0,\mathbb{H}}(\underline{x})$ (or with $\mathbb{H}$) includes N variable nodes, one for each column of $\mathbb{H}$, and M function nodes (also called, in this context, **check nodes**), one for each row. A factor node and a variable node are joined by an edge if the corresponding entry in $\mathbb{H}$ is non-vanishing. Clearly this procedure can be inverted: to any factor graph with N variable nodes and M function nodes, we can associate an $M \times N$ binary matrix $\mathbb{H}$, the **adjacency matrix** of the graph, whose non-zero entries correspond to the edges of the graph.

11.1.3 Ensembles with given degree profiles

In Chapter 9 we introduced the ensembles of factor graphs $\mathbb{D}_N(\Lambda, P)$. These have N variable nodes, and the two polynomials $\Lambda(x) = \sum_{n=0}^{\infty} \Lambda_n x^n$, $P(x) = \sum_{n=0}^{\infty} P_n x^n$

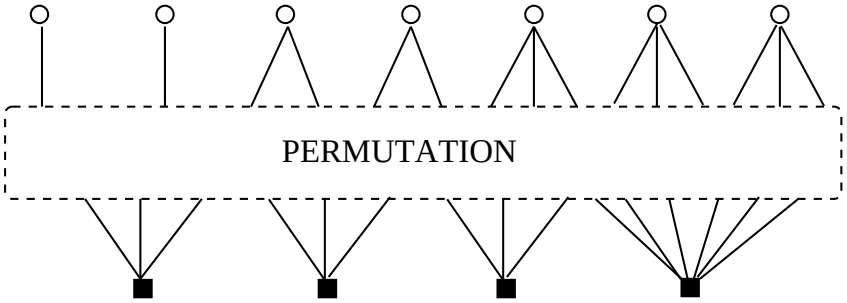

Fig. 11.1 Construction of a graph/code with given degree profiles. Here a graph with $N = 7$, $M = 4$, $\Lambda(x) = \frac{1}{7}(2x + 2x^2 + 3x^3)$, and $P(x) = \frac{1}{4}(3x^3 + x^6)$. The 'sockets' from the variable nodes and those from the checks are connected through a uniformly random permutation.

define the degree profiles: Λ_n is the probability that a randomly chosen variable node has degree n, P_n is the probability that a randomly chosen function node has degree n.

We define $\mathrm{LDPC}_N(\Lambda, P)$ to be the ensemble of linear codes whose parity check matrix is the adjacency matrix of a random graph from the $\mathbb{D}_N(\Lambda, P)$ ensemble. We will be interested in the limit $N \to \infty$ while keeping the degree profiles fixed. Therefore each vertex has bounded degree, and hence the parity check matrix has 'low density.'

In order to eliminate trivial cases, we always assume that variable nodes have degree at least 1, and function nodes at least 2. The numbers of parity check and variable nodes satisfy the relation $M = N\Lambda'(1)/P'(1)$. The ratio $L/N = (N - M)/N = 1 - \Lambda'(1)/P'(1)$, which is a lower bound to the actual rate R, is called the **design rate** R_{des} of the code (or of the ensemble). The actual rate of a code from the $\mathrm{LDPC}_N(\Lambda, P)$ ensemble is of course a random variable, but we will see below that it is in general sharply concentrated 'near' R_{des}.

A special case which is often considered is the one of 'regular' graphs: all variable nodes have degree l and all function nodes have degree k, (i.e. $P(x) = x^k$ and $\Lambda(x) = x^l$). The corresponding code ensemble is usually simply denoted as $\mathrm{LDPC}_N(l, k)$, or, more synthetically as (l, k). It has design rate $R_{\mathrm{des}} = 1 - \frac{l}{k}$.

Generating a uniformly random graph from the $\mathbb{D}_N(\Lambda, P)$ ensemble is not a trivial task. The simplest way to by-pass this problem consists in substituting the uniformly random ensemble with a slightly different one which has a simple algorithmic description. One can proceed for instance as follows. First separate the set of variable nodes uniformly at random into subsets of sizes $N\Lambda_0$, $N\Lambda_1$, ..., $N\Lambda_{l_{\max}}$, and attribute 0 'sockets' to the nodes in the first subset, 1 socket to each of the nodes in the second, and so on. Analogously, separate the set of check nodes into subsets of size MP_0, MP_1, ..., $MP_{k_{\max}}$ and attribute to nodes in each subset $0, 1, \ldots, k_{\max}$ socket. At this point the variable nodes have $N\Lambda'(1)$ sockets, and so have the check nodes. Draw a uniformly random permutation over $N\Lambda'(1)$ objects and connect the sockets on the two sides accordingly (see Fig. 11.1).

Exercise 11.1 In order to sample a graph as described above, one needs two routines. The first one separates a set of N objects uniformly into subsets of prescribed sizes. The second one samples a random permutation over a $N\Lambda'(1)$. Show that both of these tasks can be accomplished with $O(N)$ operations (having at your disposal a random number generator).

This procedure has two flaws: (*i*) It does not sample uniformly $\mathbb{D}_N(\Lambda, P)$, because two distinct factor graphs may correspond to a different number of permutations. (*ii*) It may generate multiple edges joining the same couple of nodes in the graph.

In order to cure the last problem, we shall agree that each time n edges join any two nodes, they must be erased if n is even, and they must be replaced by a single edge if n is odd. Of course the resulting graph does not necessarily have the prescribed degree profile (Λ, P), and even if we condition on this to be the case, its distribution is not uniform. We shall nevertheless insist in denoting the ensemble as $\mathrm{LDPC}_N(\Lambda, P)$. The intuition is that, for large N, the degree profile is 'close' to the prescribed one and the distribution is 'uniform enough' for our purposes. Moreover -and this is really important- this, or similar graph generation techniques are used in practice.

Exercise 11.2 This exercise aims at proving that, for large N, the degree profile produced by the explicit construction is close to the prescribed one.

(*a*) Let m be the number of multiple edges appearing in the graph and compute its expectation. Show that $\mathbb{E}\, m = O(1)$ as $N \to \infty$ with Λ and P fixed.

(*b*) Let (Λ', P') be the degree profile produced by the above procedure. Denote by

$$d \equiv \sum_l |\Lambda_l - \Lambda'_l| + \sum_k |P_k - P'_k|\,, \tag{11.3}$$

the 'distance' between the prescribed and the actual degree profiles. Derive an upper bound on d in terms of m and show that it implies $\mathbb{E}\, d = O(1/N)$.

11.2 Geometry of the codebook

As we saw in Sec. 6.2, a classical approach to the analysis of error correcting codes consists in studying the geometric properties of the corresponding codebooks. An important example of such properties is the distance enumerator $\mathcal{N}_{\underline{x}_0}(d)$, giving the number of codewords at Hamming distance d from $\underline{x}_0$. In the case of linear codes, the distance enumerator does not depend upon the reference codeword $\underline{x}_0$ (the reader is invited to prove this statement). It is therefore customary to take the all-zeros codeword as the reference, and to use the denomination **weight enumerator**: $\mathcal{N}(w) = \mathcal{N}_{\underline{x}_0}(d = w)$ is the number of codewords having **weight** (the number of ones in the codeword) equal to w.

In this section we want to estimate the expected weight enumerator $\overline{\mathcal{N}}(w) \equiv \mathbb{E}\mathcal{N}(w)$, for a random code in the $\mathrm{LDPC}_N(\Lambda, P)$ ensemble. As for the random code

ensemble of Sec. 6.2, that $\overline{\mathcal{N}}(w)$ grows exponentially in the block-length N, and that most of the codewords have a weight $w = N\omega$ growing linearly with N. We will in fact compute the exponential growth rate $\phi(\omega)$ defined by

$$\overline{\mathcal{N}}(w = N\omega) \doteq e^{N\phi(\omega)} \ . \tag{11.4}$$

In the jargon of statistical physics, $\overline{\mathcal{N}}(w)$ is an 'annealed average,' hence it may be dominated by rare instances in the ensemble. On the other hand, one expects $\log \mathcal{N}(w)$ to be tightly concentrated around its typical value $N\phi_{\rm q}(\omega)$. The typical exponent $\phi_{\rm q}(\omega)$ can be computed through a quenched calculation, for instance considering $\lim_{N\to\infty} N^{-1}\mathbb{E}\log[1+\mathcal{N}(w)]$. Of course $\phi_{\rm q}(\omega) \le \phi(\omega)$ because of the concavity of the logarithm. In this chapter we keep to the annealed calculation, which is much easier and gives an upper bound of the quenched result $\phi_{\rm q}$.

Let $\underline{x} \in \{0,1\}^N$ be a binary word of length N and weight w. Notice that $\mathbb{H}\underline{x} = 0 \mod 2$ if and only if the corresponding factor graph has the following property. Consider all the w variable nodes i such that $x_i = 1$, and color in red all edges incident on these nodes. Color in blue all the other edges. Then all the check nodes must have an even number of incident red edges. A little thought shows that $\overline{\mathcal{N}}(w)$ is the number of 'colored' factor graphs having this property for some set of w variable nodes, divided by the total number of factor graphs in the ensemble. We shall compute this number first for a graph with fixed degrees, i.e. for codes in the $\mathrm{LDPC}_N(l,k)$ ensemble, and then we shall generalize to arbitrary degree profiles.

11.2.1 Weight enumerator: regular ensembles

In the fixed degree case we have N variable nodes of degree l, M function nodes of degree k. We denote by $F = Mk = Nl$ the total number of edges. A valid colored graph must have $E = wl$ red edges. It can be constructed as follows. First choose w variable nodes, which can be done in $\binom{N}{w}$ ways. Assign to each node in this set l red sockets, and to each node outside the set l blue sockets. Then, for each of the M function nodes, color in red an even subset of its sockets in such a way that the total number of red sockets is $E = wl$. Let m_r be the number of function nodes with r red sockets. The numbers m_r can be non-zero only when r is even, and they are constrained by $\sum_{r=0}^{k} m_r = M$ and $\sum_{r=0}^{k} r m_r = lw$. The number of ways one can color the sockets of the function nodes is thus:

$$\begin{aligned}\mathcal{C}(k,M,w) = \sum_{m_0,\dots,m_k}^{(e)} & \binom{M}{m_0,\dots,m_k} \prod_r \binom{k}{r}^{m_r} \\ & \mathbb{I}\Big(\sum_{r=0}^{k} m_r = M\Big)\ \mathbb{I}\Big(\sum_{r=0}^{k} r m_r = lw\Big)\ ,\end{aligned} \tag{11.5}$$

where the sum $\sum^{(e)}$ means that non-zero m_r appear only for r even. Finally we join the variable node and check node sockets in such a way that colors are matched. There are $(lw)!(F-lw)!$ such matchings out of the total number of $F!$ corresponding to different element in the ensemble. Putting everything together, we get the final formula:

$$\overline{\mathcal{N}}(w) = \frac{(lw)!(F-lw)!}{F!} \binom{N}{w} \mathcal{C}(k,M,w) \ . \tag{11.6}$$

In order to compute the function $\phi(\omega)$ in (11.4), one needs to work out the asymptotic behavior of this formula when $N \to \infty$ at fixed $\omega = w/N$. Assuming that $m_r = x_r M = x_r N l/k$, one can expand the multinomial factors using Stirling's formula. This gives:

$$\phi(\omega) = \max_{\{x_r\}}{}^{*} \left[(1-l)\mathcal{H}(\omega) + \frac{l}{k} \sum_r \left(-x_r \log x_r + x_r \log \binom{k}{r} \right) \right] \ , \tag{11.7}$$

where the max* is taken over all choices of $x_0, x_2, x_4, \dots$ in $[0,1]$, subject to the two constraints $\sum_r x_r = 1$ and $\sum_r r x_r = k\omega$. The maximization can be done by imposing these constraints via two Lagrange multipliers. One gets $x_r = C z^r \binom{k}{r} \mathbb{I}(r \text{ even})$, where C and z are two constants fixed by the equations:

$$C = \frac{2}{(1+z)^k + (1-z)^k} \ , \tag{11.8}$$

$$\omega = z \frac{(1+z)^{k-1} - (1-z)^{k-1}}{(1+z)^k + (1-z)^k} \ . \tag{11.9}$$

Plugging back the resulting x_r into the expression (11.7) of ϕ, this gives finally:

$$\phi(\omega) = (1-l)\mathcal{H}(\omega) + \frac{l}{k} \log \frac{(1+z)^k + (1-z)^k}{2} - \omega l \log z \ , \tag{11.10}$$

where z is the function of ω defined in (11.9).

We shall see in the next sections how to use this result, but let us first explain how it can be generalized.

11.2.2 Weight enumerator: general case

We want compute the leading exponential behavior $\overline{\mathcal{N}}(w) \doteq \exp[N\phi(\omega)]$ of the expected weight enumerator for a general LDPC$_N(\Lambda, P)$ code. The idea of the approach is the same as the one we have just used for the case of regular ensembles, but the computation becomes heavier. It is therefore useful to adopt a more powerful formalism. Altogether this section is more technical than the others: the reader who is not interested in the details can skip it and go to the results.

We want to build a valid colored graph, let us denote by E its number of red edges (which is no longer fixed by w). There are $\mathsf{coeff}[\prod_l (1+xy^l)^{N\Lambda_l}, x^w y^E]$ ways of choosing the w variable nodes in such a way that their degrees add up to E [1]. As before, for each of the M function nodes, we color in red an even subset of its sockets in such a way that the total number of red sockets is E. This can be done in $\mathsf{coeff}[\prod_k q_k(z)^{MP_k}, z^E]$ ways, where $q_k(z) \equiv \frac{1}{2}(1+z)^k + \frac{1}{2}(1-z)^k$. The numbers of ways one can match the red sockets in variable and function nodes is still $E!(F-E)!$,

[1] We denote by $\mathsf{coeff}[f(x), x^n]$ the coefficient of x^n in the formal power series $f(x)$.

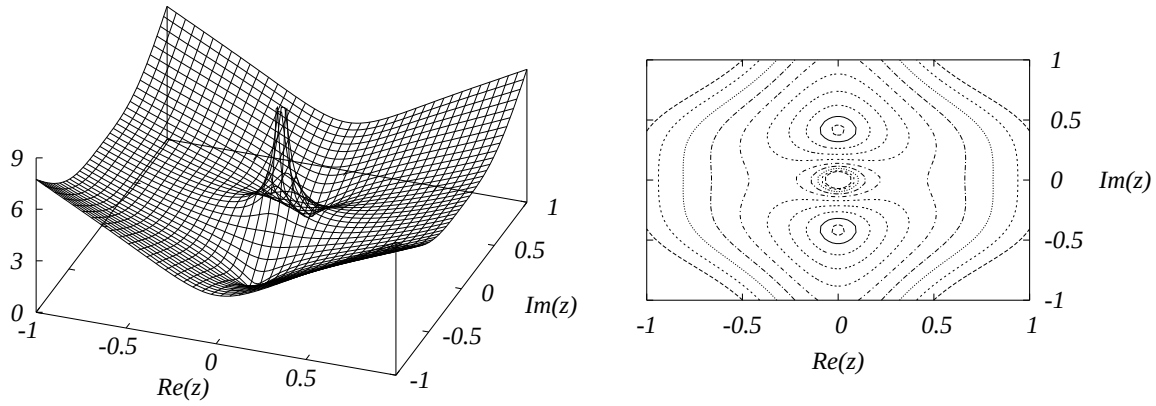

Fig. 11.2 Modulus of the function $z^{-3\xi}\, q_4(z)^{3/4}$ for $\xi = 1/3$.

where $F = N\Lambda'(1) = MP'(1)$ is the total number of edges in the graph. This gives the result:

$$\overline{\mathcal{N}}(w) = \sum_{E=0}^{F} \frac{E!(F-E)!}{F!} \qquad \mathsf{coeff}\left[\prod_{l=1}^{l_{\max}} (1+xy^l)^{N\Lambda_l}, x^w y^E\right] \mathsf{coeff}\left[\prod_{k=2}^{k_{\max}} q_k(z)^{MP_k}, z^E\right] . \quad (11.11)$$

In order to estimate the leading exponential behavior of $\overline{\mathcal{N}}(w)$ at large N, when $w = N\omega$, we set $E = F\xi = N\Lambda'(1)\xi$. The asymptotic behaviors of the $\mathsf{coeff}[\dots,\dots]$ terms can be estimated using the saddle point method. Here we sketch the idea for the second of these terms. By Cauchy theorem

$$\mathsf{coeff}\left[\prod_{k=2}^{k_{\max}} q_k(z)^{MP_k}, z^E\right] = \oint \frac{1}{z^{N\Lambda'(1)\xi+1}} \prod_{k=2}^{k_{\max}} q_k(z)^{MP_k} \frac{\mathrm{d}z}{2\pi i} \equiv \oint \frac{f(z)^N}{z} \frac{\mathrm{d}z}{2\pi i} , \quad (11.12)$$

where the integral runs over any path encircling the origin of the complex z plane in the anticlockwise direction, and

$$f(z) \equiv \frac{1}{z^{\Lambda'(1)\xi}} \prod_{k=2}^{k_{\max}} q_k(z)^{\Lambda'(1)P_k/P'(1)} . \quad (11.13)$$

In Fig. 11.2 we plot the modulus of the function $f(z)$ for degree distributions $\Lambda(x) = x^3$, $P(x) = x^4$ and $\xi = 1/3$. The function has two saddle points in $\pm z_*$, where $z_* = z_*(\xi) \in \mathbb{R}_+$ solves the equation $f'(z_*) = 0$, namely:

$$\xi = \sum_{k=2}^{k_{\max}} \rho_k \, z_* \frac{(1+z_*)^{k-1} - (1-z_*)^{k-1}}{(1+z_*)^k + (1-z_*)^k} \, . \tag{11.14}$$

Here we used the notation $\rho_k \equiv kP_k/P'(1)$ already introduced in Sec. 9.5 (analogously, we shall write $\lambda_l \equiv l\Lambda_l/\Lambda'(1)$). This equation generalizes (11.9). If we take the integration contour in Eq. (11.12) to be the circle of radius z_*, the integral is dominated by the saddle point at z_* (together with the symmetric point $-z_*$). We get therefore

$$\mathsf{coeff}\left[\prod_{k=2}^{k_{\max}} q_k(z)^{MP_k}, z^E\right] \doteq \exp\left\{N\left[-\Lambda'(1)\xi\log z_* + \frac{\Lambda'(1)}{P'(1)}\sum_{k=2}^{k_{\max}} P_k \log q_k(z_*)\right]\right\} .$$

Proceeding analogously with the second $\mathsf{coeff}[\dots,\dots]$ term in Eq. (11.11), we get $\overline{\mathcal{N}}(w = N\omega) \doteq \exp\{N\phi(\omega)\}$. The function ϕ is given by

$$\phi(\omega) = \sup_{\xi} \inf_{x,y,z} \left\{-\Lambda'(1)\mathcal{H}(\xi) - \omega\log x - \Lambda'(1)\xi\log(yz) + \right. \\ \left. + \sum_{l=2}^{l_{\max}} \Lambda_l \log(1+xy^l) + \frac{\Lambda'(1)}{P'(1)}\sum_{k=2}^{k_{\max}} P_k \log q_k(z)\right\} , \tag{11.15}$$

where the minimization over x, y, z is understood to be taken over the positive real axis while $\xi \in [0,1]$. The stationarity condition with respect to variations of z is given by Eq. (11.14). Stationarity with respect to ξ, x, y yields, respectively

$$\xi = \frac{yz}{1+yz} \, , \qquad \omega = \sum_{l=1}^{l_{\max}} \Lambda_l \frac{xy^l}{1+xy^l} \, , \qquad \xi = \sum_{l=1}^{l_{\max}} \lambda_l \frac{xy^l}{1+xy^l} \, . \tag{11.16}$$

If we use the first of these equations to eliminate ξ, we obtain the final parametric representation (in the parameter $x \in [0,\infty[$) of $\phi(\omega)$:

$$\phi(\omega) = -\omega\log x - \Lambda'(1)\log(1+yz) + \sum_{l=1}^{l_{\max}} \Lambda_l \log(1+xy^l) + \\ + \frac{\Lambda'(1)}{P'(1)}\sum_{k=2}^{k_{\max}} P_k \log q_k(z) \, , \tag{11.17}$$

$$\omega = \sum_{l=1}^{l_{\max}} \Lambda_l \frac{xy^l}{1+xy^l} \, . \tag{11.18}$$

The two functions $y = y(x)$ and $z = z(x)$ are solutions of the coupled equations

$$y = \frac{\sum_{k=2}^{k_{\max}} \rho_k \, p_k^-(z)}{\sum_{k=2}^{k_{\max}} \rho_k \, p_k^+(z)} \, , \qquad z = \frac{\sum_{l=1}^{l_{\max}} \lambda_l x y^{l-1}/(1+xy^l)}{\sum_{l=1}^{l_{\max}} \lambda_l/(1+xy^l)} \, , \tag{11.19}$$

where we defined $p_k^{\pm}(z) \equiv \frac{(1+z)^{k-1} \pm (1-z)^{k-1}}{(1+z)^k + (1-z)^k}$.

Exercise 11.3 The numerical solution of Eqs. (11.18) and (11.19) can be somewhat tricky. Here is a simple iterative procedure which usually works reasonably well. The reader is invited to try it with her favorite degree distributions Λ, P.

First, solve Eq. (11.18) for x at given $y \in [0, \infty[$ and $\omega \in [0, 1]$, using a bisection method. Next, substitute this value of x in Eq. (11.19), and write the resulting equations as $y = f(z)$ and $z = g(y, \omega)$. Define $F_\omega(y) \equiv f(g(y, \omega))$. Solve the equation $y = F_\omega(y)$ by iteration of the map $y_{n+1} = F_\omega(y_n)$ Once the fixed point y_* is found, the other parameters are computed as $z_* = g(y_*, \omega)$ and x_* is the solution of Eq. (11.18) for $y = y_*$. Finally x_*, y_*, z_* are substituted in Eq. (11.17) to obtain $\phi(\omega)$.

Examples of functions $\phi(\omega)$ are shown in Figures 11.3, 11.4, 11.5. We shall now discuss these results, paying special attention to the region of small ω.

11.2.3 Short distance properties

In the low noise limit, the performance of a code depends a lot on the existence of codewords at short distance from the transmitted one. For linear codes and symmetric communication channels, we can assume without loss of generality that the all zeros codeword has been transmitted. Here we will work out the short distance (i.e. small weight ω) behavior of $\phi(\omega)$ for several LDPC ensembles. These properties will be used to characterize the code performances in Sect. 11.3.

As $\omega \to 0$, solving Eqs. (11.18) and (11.19) yields $y, z \to 0$. By Taylor expansion of these equations, we get

$$y \simeq \rho'(1) z \,, \qquad z \simeq \lambda_{l_{\min}} x y^{l_{\min}-1} \,, \qquad \omega \simeq \Lambda_{l_{\min}} x y^{l_{\min}} \,, \tag{11.20}$$

where we neglected higher order terms in y, z. At this point we must distinguish whether $l_{\min} = 1$, $l_{\min} = 2$ or $l_{\min} \geq 3$.

We start with the case $l_{\min} = 1$. Then x, y, z all scale like $\sqrt{\omega}$, and a short computation shows that

$$\phi(\omega) = -\frac{1}{2}\, \omega \, \log\left(\omega/\Lambda_1^2\right) + O(\omega) \,. \tag{11.21}$$

In particular $\phi(\omega)$ is strictly positive for ω sufficiently small. The expected number of codewords within a small relative Hamming distance $w = N\omega$ from a given codeword is exponential in N. Furthermore, Eq. (11.21) is reminiscent of the behavior in absence of any parity check, where one gets $\phi(\omega) = \mathcal{H}(\omega) \simeq -\omega \log \omega$.

Exercise 11.4 In order to check Eq. (11.21), compute the weight enumerator for the regular LDPC$_N(l = 1, k)$ ensemble. Notice that, in this case the weight enumerator does not depend on the code realization and admits the simple representation $\mathcal{N}(w) = \mathsf{coeff}[q_k(z)^{N/k}, z^w]$.

An example of weight enumerator for an irregular code with $l_{\min} = 1$ is shown in Fig. 11.3. The behavior (11.21) is quite bad for an error correcting code. In order to

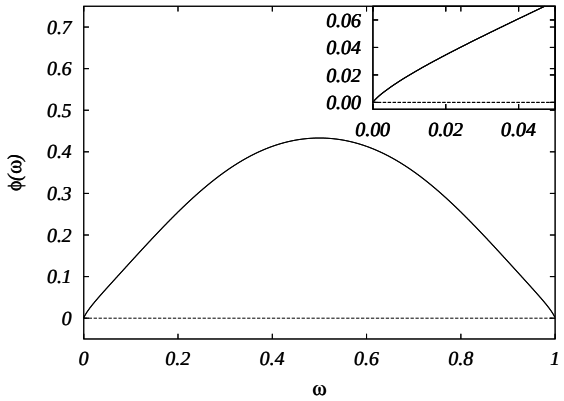

Fig. 11.3 Logarithm of the expected weight enumerator, $\phi(\omega)$, plotted versus the reduced weight $\omega = w/N$, for the ensemble $\mathrm{LDPC}_N(\frac{1}{4}x + \frac{1}{4}x^2 + \frac{1}{2}x^3, x^6)$. Inset: small weight region. $\phi(\omega)$ is positive near to the origin, and in fact its derivative diverges as $\omega \to 0$: each codeword is surrounded by a large number of very close other codewords. This makes it a bad error correcting code.

understand why, let us for a moment forget that this result was obtained by taking $\omega \to 0$ *after* $N \to \infty$, and apply it in the regime $N \to \infty$ at $w = N\omega$ fixed. We get

$$\overline{\mathcal{N}}(w) \sim \left(\frac{N}{w}\right)^{\frac{1}{2}w} . \tag{11.22}$$

It turns out that this result holds not only in average but for most codes in the ensemble. In other words, already at Hamming distance 2 from any given codeword there are $\Theta(N)$ other codewords. It is intuitively clear that discriminating between two codewords at $\Theta(1)$ Hamming distance, given a noisy observation, is in most of the cases impossible. Because of these remarks, one usually discards $l_{\min} = 1$ ensembles in error correction.

Consider now the case $l_{\min} = 2$. From Eq. (11.20), we get

$$\phi(\omega) \simeq A\,\omega\,, \qquad A \equiv \log\left[\frac{P''(1)}{P'(1)}\frac{2\Lambda_2}{\Lambda'(1)}\right] = \log\left[\lambda'(0)\rho'(1)\right] . \tag{11.23}$$

As it will appear in Ch. 15, the combination $\lambda'(0)\rho'(1)$ has an important concrete interpretation.

The code ensemble has significantly different properties depending on the sign of A. If $A > 0$, the expected number of codewords within a small (but $\Theta(N)$) Hamming distance from any given codeword is exponential in the block-length. The situation seems similar to the $l_{\min} = 1$ case. Notice however that $\phi(\omega)$ goes much more quickly to 0 as $\omega \to 0$ in the present case. Assuming again that (11.23) holds beyond the asymptotic regime in which it was derived, we get

$$\overline{\mathcal{N}}(w) \sim e^{Aw} . \tag{11.24}$$

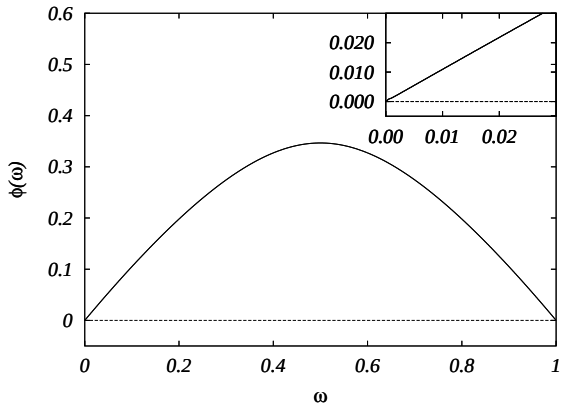

Fig. 11.4 Logarithm of the expected weight enumerator for the $\mathrm{LDPC}_N(2,4)$ ensemble. The degree profiles are $\Lambda(x) = x^2$, meaning that all variable nodes have degree 2, and $P(x) = x^4$, meaning that all function nodes have degree 4. Inset: small weight region. The constant A is positive, so there exist codewords at short distances.

In other words the number of codewords around any particular one is $o(N)$ until we reach a Hamming distance $d_* \simeq \log N/A$. For many purposes d_* plays the role of an 'effective' minimum distance. The example of the regular code $\mathrm{LDPC}_N(2,4)$, for which $A = \log 3$, is shown in Fig. 11.4

If on the other hand $A < 0$, then $\phi(\omega) < 0$ in some interval $\omega \in]0, \omega_*[$. The first moment method then shows that there are no codewords of weight 'close to' $N\omega$ for any ω in this range.

A similar conclusion is reached if $l_{\min} \geq 3$, where one finds:

$$\phi(\omega) \simeq \left(\frac{l_{\min}-2}{2}\right) \omega \log\left(\frac{\omega}{\Lambda_{l_{\min}}}\right) , \tag{11.25}$$

An example of weight enumerator exponent for a code with good short distance properties, the $\mathrm{LDPC}_N(3,6)$ code, is given in Fig. 11.5.

This discussion can be summarized as:

Proposition 11.1 *Consider a random linear code from the* $\mathrm{LDPC}_N(\Lambda, P)$ *ensemble with* $l_{\min} \geq 3$. *Let* $\omega_* \in]0, 1/2[$ *be the first non-trivial zero of* $\phi(\omega)$, *and consider any interval* $[\omega_1, \omega_2] \subset]0, \omega_*[$. *With high probability, there does not exist any pair of codewords with distance belonging to this interval. The same result holds when* $l_{\min} = 2$ *and* $\lambda'(0)\rho'(1) = \frac{P''(1)}{P'(1)} \frac{2\Lambda_2}{\Lambda'(1)} < 1$.

Notice that our study only deals with weights $w = \omega N$ which grow linearly with N. The proposition excludes the existence of codewords of arbitrarily small ω, but it does not tell anything about possible codewords of sub-linear weight: $w = o(N)$ (for instance, with w finite as $N \to \infty$). It turns out that, if $l_{\min} \geq 3$, the code has with high probability no such codewords, and its minimum distance is at least $N\omega_*$. If on the other hand $l_{\min} = 2$, the code has typically some codewords of finite weight w.

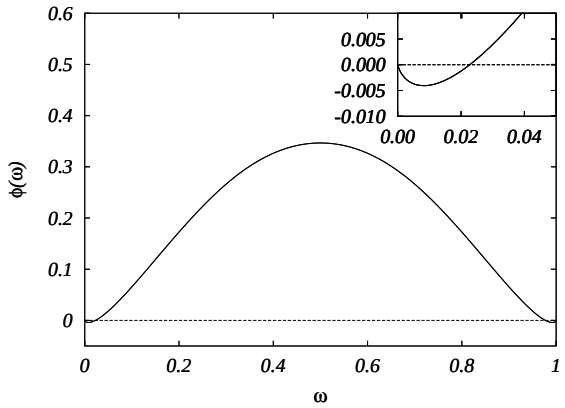

Fig. 11.5 Logarithm of the expected weight enumerator for the $\mathrm{LDPC}_N(3,6)$ ensemble. Inset: small weight region. $\phi(\omega) < 0$ for $\omega < \omega_* \approx 0.02$. There are no codewords except from the 'all-zeros' one in the region $\omega < \omega_*$.

However (if $A < 0$), they can be eliminated without changing the code rate by an 'expurgation' procedure similar to the one described in Sec. 6.5.1.

11.2.4 Rate

The weight enumerator can also be used to obtain a precise characterization of the rate of a $\mathrm{LDPC}_N(\Lambda, P)$ code. For $\omega = 1/2$, $x = y = z = 1$ satisfy Eqs. (11.18) and (11.19). This gives:

$$\phi(\omega = 1/2) = \left(1 - \frac{\Lambda'(1)}{P'(1)}\right) \log 2 = R_{\mathrm{des}} \log 2 \,. \tag{11.26}$$

It turns out that, in most cases of practical interest, the curve $\phi(\omega)$ has its maximum at $\omega = 1/2$ (see for instance the figures 11.3, 11.4, 11.5). In such cases the result (11.26) shows that the rate equals the design rate:

Proposition 11.2 *Let R be the rate of a code from the $\mathrm{LDPC}_N(\Lambda, P)$ ensemble, $R_{\mathrm{des}} = 1 - \Lambda'(1)/P'(1)$ the associated design rate and $\phi(\omega)$ the function defined in Eqs. (11.17) to (11.19). Assume that $\phi(\omega)$ achieves its absolute maximum over the interval $[0,1]$ at $\omega = 1/2$. Then, for any $\delta > 0$, there exists a positive N-independent constant $C_1(\delta)$ such that*

$$\mathbb{P}\{|R - R_{\mathrm{des}}| \geq \delta\} \leq C_1(\delta)\, 2^{-N\delta/2} \,. \tag{11.27}$$

Proof: Since we already established that $R \geq R_{\mathrm{des}}$, we only need to prove an upper bound on R. The rate is defined as $R \equiv (\log_2 \mathcal{N})/N$, where $\mathcal{N}$ is the total number of codewords. Markov's inequality gives:

$$\mathbb{P}\{R \geq R_{\mathrm{des}} + \delta\} = \mathbb{P}\{\mathcal{N} \geq 2^{N(R_{\mathrm{des}}+\delta)}\} \leq 2^{-N(R_{\mathrm{des}}+\delta)}\, \mathbb{E}\mathcal{N} \,. \tag{11.28}$$

The expectation of the number of codewords is $\mathbb{E}\mathcal{N}(w) \doteq \exp\{N\phi(w/N)\}$, and there are only $N+1$ possible values of the weight w, therefore:

$$\mathbb{E}\mathcal{N} \doteq \exp\{N \sup_{\omega\in[0,1]} \phi(\omega)\}\,, \tag{11.29}$$

As $\sup\phi(\omega) = \phi(1/2) = R_{\rm des}\log 2$ by hypothesis, there exists a constant $C_1(\delta)$ such that, for any N, $\mathbb{E}\mathcal{N} \le C_1(\delta)2^{N(R_{\rm des}+\delta/2)}$ for any N. Plugging this into Eq. (11.28), we get

$$\mathbb{P}\{R \ge R_{\rm des}+\delta\} \le C_1(\delta)\,2^{-N\delta/2}\,. \tag{11.30}$$

□

11.3 LDPC codes for the binary symmetric channel

Our study of the weight enumerator has shown that codes from the $\mathrm{LDPC}_N(\Lambda, P)$ ensemble with $l_{\rm min} \ge 3$ have a good short distance behavior. The absence of codewords within an extensive distance $N\omega_*$ from the transmitted one guarantees that any error (even introduced by an adversarial channel) changing a fraction of the bits smaller than $\omega_*/2$ can be corrected. Here we want to study the performance of these codes in correcting *typical* errors introduced from a given (probabilistic) channel. We will focus on the BSC(p) which flips each bit independently with probability $p < 1/2$. Supposing as usual that the all-zero codeword $\underline{x}^{(0)} = \underline{0}$ has been transmitted, let us call $\underline{y} = (y_1 \dots y_N)$ the received message. Its components are i.i.d. random variables taking value 0 with probability $1-p$, value 1 with probability p. The decoding strategy which minimizes the block error rate is word MAP (or maximum likelihood) decoding, which outputs the codeword closest to the channel output $\underline{y}$. As already mentioned, we don't bother about the practical implementation of this strategy and its computational complexity.

The block error probability for a code $\mathfrak{C}$, denoted by $\mathrm{P_B}(\mathfrak{C})$, is the probability that there exists a 'wrong' codeword, distinct from $\underline{0}$, whose distance to $\underline{y}$ is smaller than $d(\underline{0},\underline{y})$. Its expectation value over the code ensemble, $\mathrm{P_B} = \mathbb{E}\,\mathrm{P_B}(\mathfrak{C})$, is an important indicator of ensemble performances. We will show that in the large N limit, codes with $l_{\rm min} \ge 3$ undergo a phase transition, separating a low noise phase, $p < p_{\rm MAP}$, in which $\lim_{N\to\infty}\mathrm{P_B}$ is zero, from a high noise phase, $p > p_{\rm MAP}$, where the limit is not zero. While the computation of $p_{\rm MAP}$ is deferred to Ch. 15, we derive here rigorous bounds which imply that appropriate LDPC codes have very good performances, close to Shannon's information theoretic limit, under MAP decoding.

11.3.1 Lower bound on the error

We start by deriving a general bound on the block error probability $\mathrm{P_B}(\mathfrak{C})$ on the BSC(p) channel, valid for any linear code. Let $\mathcal{N} = 2^{NR}$ be the size of the codebook $\mathfrak{C}$. By union bound:

$$\begin{aligned}\mathrm{P_B}(\mathfrak{C}) &= \mathbb{P}\left\{\exists\,\alpha\neq 0 \quad \text{s.t.} \quad d(\underline{x}^{(\alpha)},\underline{y}) \le d(\underline{0},\underline{y})\right\} \\ &\le \sum_{\alpha=1}^{\mathcal{N}-1} \mathbb{P}\left\{d(\underline{x}^{(\alpha)},\underline{y}) \le d(\underline{0},\underline{y})\right\}\,. \end{aligned} \tag{11.31}$$

As the components of $\underline{y}$ are i.i.d. Bernoulli variables, the probability $\mathbb{P}\{d(\underline{x}^{(\alpha)}, \underline{y}) \leq d(\underline{0}, \underline{y})\}$ depends on the vector $\underline{x}^{(\alpha)}$ only through its weight w. Let $\underline{x}(w)$ be the vector formed by w ones followed by $N-w$ zeroes, and denote by $\mathcal{N}(w)$ the weight enumerator of the code $\mathfrak{C}$. Then

$$\mathrm{P_B}(\mathfrak{C}) \leq \sum_{w=1}^{N} \mathcal{N}(w)\ \mathbb{P}\left\{d(\underline{x}(w), \underline{y}) \leq d(\underline{0}, \underline{y})\right\}\ . \tag{11.32}$$

The probability $\mathbb{P}\left\{d(\underline{x}(w), \underline{y}) \leq d(\underline{0}, \underline{y})\right\}$ can be written as $\sum_u \binom{w}{u} p^u (1-p)^{w-u}\ \mathbb{I}(u \geq w/2)$, where u is the number of sites $i \in \{1, \cdots, w\}$ such that $y_i = \mathtt{1}$. A good bound is provided by a standard method known as **Chernoff bound**.

Exercise 11.5 Let X be a random variable. Show that, for any a and any $\lambda > 0$:

$$\mathbb{P}(X \geq a) \leq e^{-\lambda a}\ \mathbb{E}\left(e^{\lambda X}\right)\ . \tag{11.33}$$

In our case this gives

$$\mathbb{P}\left\{d(\underline{x}(w), \underline{y}) \leq d(\underline{0}, \underline{y})\right\} \leq \mathbb{E} e^{\lambda[d(\underline{0},\underline{y}) - d(\underline{x}(w),\underline{y})]} = [(1-p)\, e^{-\lambda} + p\, e^{\lambda}]^w\, .$$

The bound is optimized for $\lambda = \frac{1}{2}\log(\frac{1-p}{p}) > 0$, and gives

$$\mathrm{P_B}(\mathfrak{C}) \leq \sum_{w=1}^{N} \mathcal{N}(w)\ e^{-\gamma w}\, . \tag{11.34}$$

where $\gamma \equiv -\log\sqrt{4p(1-p)} \geq 0$. The quantity $\sqrt{4p(1-p)}$ is sometimes referred to as **Bhattacharya parameter** of the channel BSC(p).

Exercise 11.6 Consider the case of a general binary memoryless symmetric channel with transition probability $Q(y|x)$, $x \in \{\mathtt{0}, \mathtt{1}\}$ $y \in \mathcal{Y} \subseteq \mathbb{R}$. First show that Eq. (11.31) remains valid if the Hamming distance $d(\underline{x}, \underline{y})$ is replaced by the log-likelihood

$$d_Q(\underline{x}|\underline{y}) = -\sum_{i=1}^{N} \log Q(y_i|x_i)\, . \tag{11.35}$$

[Hint: remember the general expressions (6.5) for the probability $\mu_y(\underline{x}) = \mathbb{P}(\underline{x}|\underline{y})$ that the transmitted codeword was $\underline{x}$, given that the received message is $\underline{y}$]. Then repeat the derivation from Eq. (11.31) to Eq. (11.34). The final expression involves $\gamma = -\log B_Q$, where the Bhattacharya parameter is defined as $B_Q = \sum_y \sqrt{Q(y|\mathtt{1})Q(y|\mathtt{0})}$.

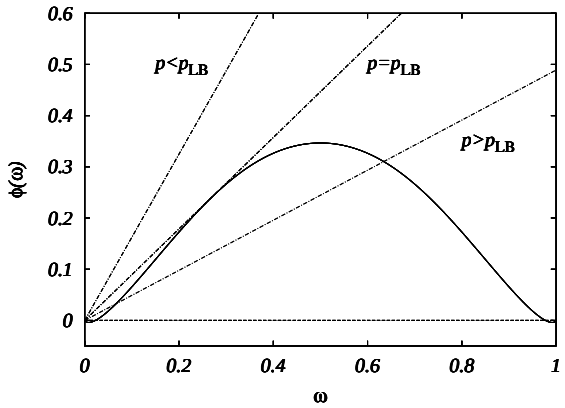

Fig. 11.6 Geometric construction yielding the lower bound on the threshold for reliable communication for the $\mathrm{LDPC}_N(3,6)$ ensemble used over the binary symmetric channel. In this case $p_{\mathrm{LB}} \approx 0.0438737$. The other two lines refer to $p = 0.01 < p_{\mathrm{LB}}$ and $p = 0.10 > p_{\mathrm{LB}}$.

Equation (11.34) shows that the block error probability depends on two factors: one is the weight enumerator, the second one, $\exp(-\gamma w)$ is a channel-dependent term: as the weight of the codewords increases, their contribution is scaled down by an exponential factor because it is less likely that the received message $\underline{y}$ will be closer to a codeword of large weight than to the all-zero codeword.

So far the discussion is valid for any given code. Let us now consider the average over $\mathrm{LDPC}_N(\Lambda, P)$ code ensembles. A direct averaging gives the bound:

$$\mathrm{P_B} \equiv \mathbb{E}_{\mathfrak{C}} \mathrm{P_B}(\mathfrak{C}) \leq \sum_{w=1}^{N} \overline{\mathcal{N}}(w)\, e^{-\gamma w} \doteq \exp\left\{ N \sup_{\omega \in]0,1]} [\phi(\omega) - \gamma\omega] \right\} . \qquad (11.36)$$

As such, this expression is useless, because the $\sup_\omega[\phi(\omega) - \gamma\omega]$, being larger or equal than the value at $\omega = 0$, is always positive. However, if we restrict to ensembles with $l_{\min} \geq 3$, we know that, with probability going to one in the large N limit, there exists no codeword in the ω interval $]0, \omega_*[$. In such cases, the maximization over ω in (11.36) can be performed in the interval $[\omega_*, 1]$ instead of $]0, 1]$. (By Markov inequality, this is true whenever $N \sum_{w=1}^{N\omega_*-1} \overline{\mathcal{N}}(w) \to 0$ as $N \to \infty$). The bound becomes useful whenever the supremum $\sup_{\omega \in [\omega_*, 1]}[\phi(\omega) - \gamma\omega] < 0$: then $\mathrm{P_B}$ vanishes in the large N limit. We have thus obtained:

Proposition 11.3 *Consider the average block error rate* $\mathrm{P_B}$ *for a random code in the* $\mathrm{LDPC}_N(\Lambda, P)$ *ensemble, with* $l_{\min} \geq 3$*, used over a BSC*(p) *channel, with* $p < 1/2$*. Let* $\gamma \equiv -\log\sqrt{4p(1-p)}$ *and let* $\phi(\omega)$ *be the the weight enumerator exponent, defined in (11.4) [*$\phi(\omega)$ *can be computed using Eqs. (11.17), (11.18), and (11.19)]. If* $\phi(\omega) < \gamma\omega$ *for any* $\omega \in (0, 1]$ *such that* $\phi(\omega) \geq 0$*, then* $\mathrm{P_B} \to 0$ *in the large block-length limit.*

This result has a pleasing geometric interpretation which is illustrated in Fig. 11.6. As p increases from 0 to 1/2, γ decreases from $+\infty$ to 0. The condition $\phi(\omega) < \gamma\omega$

can be rephrased by saying that the weight enumerator exponent $\phi(\omega)$ must lie below the straight line of slope γ through the origin. Let us call $p_{\rm LB}$ the smallest value of p such that the line $\gamma\omega$ touches $\phi(\omega)$.

The geometric construction implies $p_{\rm LB} > 0$. Furthermore, for p large enough Shannon's theorem implies that $\mathrm{P_B}$ is bounded away from 0 for any non-vanishing rate $R > 0$. The **MAP threshold** $p_{\rm MAP}$ for the ensemble $\mathrm{LDPC}_N(\Lambda, P)$ can be defined as the largest (or, more precisely, the supremum) value of p such that $\lim_{N\to\infty} \mathrm{P_B} = 0$. This definition has a very concrete practical meaning: for any $p < p_{\rm MAP}$ one can communicate with an arbitrarily small error probability, by using a code from the $\mathrm{LDPC}_N(\Lambda, P)$ ensemble provided N is large enough. Proposition 11.3 then implies:

$$p_{\rm MAP} \geq p_{\rm LB}\,. \tag{11.37}$$

In general one expects $\lim_{N\to\infty} \mathrm{P_B}$ to exist (and to be strictly positive) for $p > p_{\rm MAP}$. However, there exists no proof of this statement.

It is interesting to notice that, at $p = p_{\rm LB}$, our upper bound on $\mathrm{P_B}$ is dominated by codewords of weight $w \approx N\tilde{\omega}$, where $\tilde{\omega} > 0$ is the value where $\phi(\omega) - \gamma\omega$ is maximum. This suggests that, each time an error occurs, a finite fraction of the bits are decoded incorrectly and this fraction fluctuates little from transmission to transmission (or, from code to code in the ensemble). The geometric construction also suggests the less obvious (but essentially correct) guess that this fraction jumps discontinuously from 0 to a finite value when p crosses the critical value $p_{\rm MAP}$. Finally $\tilde{\omega} > \omega_*$ strictly: dominant error events are not triggered by the closest codewords!

Exercise 11.7 Let us study the case $l_{\rm min} = 2$. Proposition 11.3 is no longer valid, but we can still apply Eq. (11.36).

(*a*) Consider the $(2, 4)$ ensemble whose weight enumerator exponent is plotted in Fig. 11.4, the small weight behavior being given by Eq. (11.24). At small enough p, it is reasonable to assume that the block error rate is dominated by small weight codewords. Estimate $\mathrm{P_B}$ using Eq. (11.36) under this assumption.

(*b*) Show that the assumption breaks down for $p \geq p_{\rm loc}$, where $p_{\rm loc} \leq 1/2$ solves the equation $3\sqrt{4p(1-p)} = 1$.

(*c*) Discuss the case of a general code ensemble with $l_{\rm min} = 2$, and $\phi(\omega)$ concave for $\omega \in [0, 1]$. Draw a weight enumerator exponent $\phi(\omega)$ such that the assumption of low-weight codewords dominance breaks down before $p_{\rm loc}$. What do you expect of the average bit error rate $\mathrm{P_b}$ for $p < p_{\rm loc}$? And for $p > p_{\rm loc}$?

Exercise 11.8 Discuss the qualitative behavior of the block error rate for the cases where $l_{\rm min} = 1$.

11.3.2 Upper bound on the error

Let us consider as before the communication over a BSC(p), keeping for simplicity to regular codes LDPC$_N(l,k)$. Gallager has proved the following bound:

Theorem 11.4 *Let $p_{\rm MAP}$ be the threshold for reliable communication over the binary symmetric channel using codes from the* LDPC$_N(l,k)$*, with design rate $R_{\rm des}=1-k/l$. Then $p_{\rm MAP}\le p_{\rm UB}$, where $p_{\rm UB}\le 1/2$ is the solution of*

$$\mathcal{H}(p)=(1-R_{\rm des})\mathcal{H}\left(\frac{1-(1-2p)^k}{2}\right)\,, \tag{11.38}$$

We shall not give a full proof of this result, but we show in this section a sequence of heuristic arguments which can be turned into a proof. The details can be found in the original literature.

Assume that the all-zero codeword $\underline{0}$ has been transmitted and that a noisy vector $\underline{y}$ has been received. The receiver will look for a vector $\underline{x}$ at Hamming distance about Np from $\underline{y}$, and satisfying all the parity check equations. In other words, let us denote by $\underline{z}=\mathbb{H}\underline{x}$, $\underline{z}\in\{0,1\}^M$ (here $\mathbb{H}$ is the parity check matrix and multiplication is performed modulo 2), the **syndrome**. This is a vector with M components. If $\underline{x}$ is a codeword, all parity checks are satisfied, and we have $\underline{z}=\underline{0}$. There is at least one vector $\underline{x}$ fulfilling the conditions $d(\underline{x},\underline{y})\approx Np$, and $\underline{z}=\underline{0}$: the transmitted codeword $\underline{0}$. Decoding is successful only if it is the unique such vector.

The number of vectors $\underline{x}$ whose Hamming distance from $\underline{y}$ is close to Np is approximatively $2^{N\mathcal{H}(p)}$. Let us now estimate the number of distinct syndromes $\underline{z}=\mathbb{H}\underline{x}$, when $\underline{x}$ is on the sphere $d(\underline{x},\underline{y})\approx Np$. Writing $\underline{x}=\underline{y}\oplus\underline{x}'$, this is equivalent to counting the number of distinct vectors $\underline{z}'=\mathbb{H}\underline{x}'$ when the weight of $\underline{x}'$ is about Np. It is convenient to think of $\underline{x}'$ as a vector of N i.i.d. Bernoulli variables of mean p: we are then interested in the number of distinct *typical* vectors $\underline{z}'$. Notice that, since the code is regular, each entry z'_i is a Bernoulli variable of parameter

$$p_k=\sum_{n\ \rm odd}^{k}\binom{k}{n}p^n(1-p)^{k-n}=\frac{1-(1-2p)^k}{2}\,. \tag{11.39}$$

If the bits of $\underline{z}'$ were independent, the number of typical vectors $\underline{z}'$ would be $2^{N(1-R_{\rm des})\mathcal{H}(p_k)}$ (the dimension of $\underline{z}'$ being $M=N(1-R_{\rm des})$). It turns out that correlations between the bits decrease this number, so we can use the i.i.d. estimate to get an upper bound.

Let us now assume that for each $\underline{z}$ in this set, the number of reciprocal images (i.e. of vectors $\underline{x}$ such that $\underline{z}=\mathbb{H}\underline{x}$) is approximatively the same. If $2^{N\mathcal{H}(p)}\gg 2^{N(1-R_{\rm des})\mathcal{H}(p_k)}$, for each $\underline{z}$ there is an exponential number of vectors $\underline{x}$, such that $\underline{z}=\mathbb{H}\underline{x}$. This will be true, in particular, for $\underline{z}=\underline{0}$: the received message is therefore not uniquely decodable. In the alternative situation most of the vectors $\underline{z}$ correspond to (at most) a single $\underline{x}$. This will be the case for $\underline{z}=\underline{0}$: decoding can be successful.

11.3.3 Summary of the bounds

In Table 11.1 we consider a few regular LDPC$_N(\Lambda,P)$ ensembles over the BSC(p) channel. We show the window of possible values of the noise threshold $p_{\rm MAP}$, using

l	k	$R_{\rm des}$	LB of Sec. 11.3.1	Gallager UB	Gallager LB	Shannon limit
3	4	1/4	0.1333161	0.2109164	0.2050273	0.2145018
3	5	2/5	0.0704762	0.1397479	0.1298318	0.1461024
3	6	1/2	0.0438737	0.1024544	0.0914755	0.1100279
4	6	1/3	0.1642459	0.1726268	0.1709876	0.1739524
5	10	1/2	0.0448857	0.1091612	0.1081884	0.1100279

Table 11.1 Bounds on the threshold for reliable communication over the BSC(p) channel using LDPC$_N(l,k)$ ensembles with MAP decoding. The fourth and fifth columns are the lower bound of Proposition 11.3 and the upper bound of Theorem 11.4. The sixth column is an improved lower bound by Gallager.

the lower bound of Proposition 11.3 and the upper bound of Theorem 11.4. In most cases, the comparison is not satisfactory (the gap between upper and lower bound is close to a factor 2). A much smaller uncertainty is achieved using an improved lower bound again derived by Gallager, based on a refinement of the arguments in the previous section. As we shall see in Ch. 15 by computing $p_{\rm MAP}$, neither of the bounds is tight. On the other hand they are sufficiently good to show that, for large k, l the MAP threshold of these ensembles is close to Shannon capacity (although bounded away from it). Indeed, studying the asymptotic behavior of these bounds, one can show that the MAP threshold of the (k,l) ensemble converges to $p_{\rm Sh}$ as $k, l \to \infty$ with a fixed ratio l/k.

Exercise 11.9 Let $p_{\rm Sh}$ be the upper bound on $p_{\rm MAP}$ provided by Shannon channel coding theorem. Explicitly, $p_{\rm Sh} \le 1/2$ is the solution of $\mathcal{H}(p) = 1 - R$. Prove that, if $R = R_{\rm des}$ (as is the case with high probability for LDPC$_N(l,k)$ ensembles), then $p_{\rm UB} < p_{\rm Sh}$.

11.4 A simple decoder: bit flipping

So far we have analyzed the behavior of LDPC ensembles under the optimal (word MAP) decoding strategy. However there is no known way of implementing this decoder with an efficient algorithm. The naive algorithm goes through each codeword $\underline{x}^{(\alpha)}$, $\alpha = 0, \dots 2^{NR} - 1$ and outputs the one of greatest likelihood $Q(\underline{y}|\underline{x}^{(\alpha)})$. However this approach takes a time which grows exponentially with the block-length N. For large N (which is the regime where the error rate becomes close to optimal), this is unpractical.

LDPC codes are interesting because there exist fast sub-optimal decoding algorithms with performances close to the theoretical optimal performance, and therefore close to Shannon's limit. Here we show one example of a very simple decoding method, called the **bit flipping** algorithm. After transmission through a BSC channel, we have received the message $\underline{y}$ and try to find the sent codeword $\underline{x}$ by:

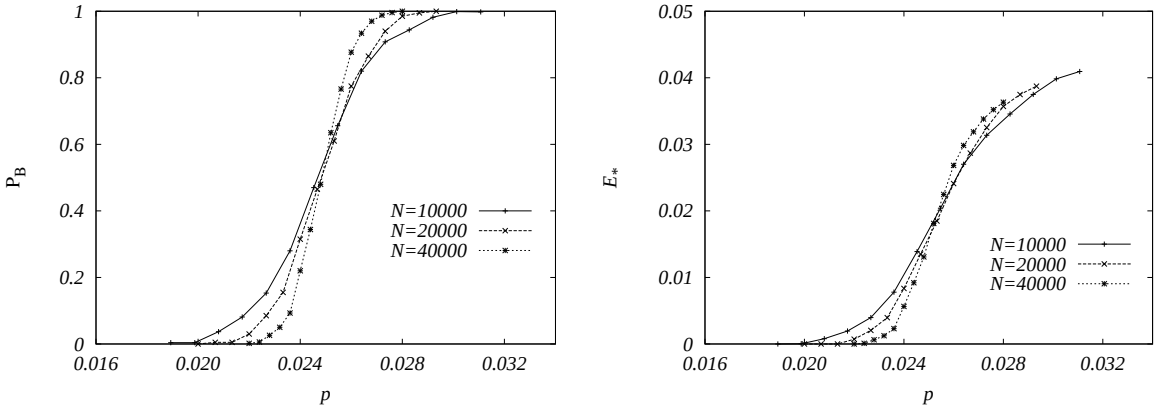

Fig. 11.7 Performances of the bit-flipping decoding algorithm on random codes from the $(5, 10)$ regular LDPC ensemble, used over the BSC(p) channel. Left: block error rate. Right: residual number of unsatisfied parity checks after the algorithm has halted. Statistical error bars are smaller than symbols.

BIT-FLIPPING DECODER (Received message $\underline{y}$)	
1:	Set $\underline{x}(0) = \underline{y}$.
2:	**for** $t = 1, \ldots, N$:
3:	find a bit belonging to more unsatisfied than satisfied parity checks;
4:	if such a bit exists, flip it: $x_i(t+1) = x_i(t) \oplus \mathtt{1}$, and keep the other bits: $x_j(t+1) = x_j(t)$ for all $j \neq i$;
5:	if there is no such bit, **return** $\underline{x}(t)$ and halt.

The bit to be flipped is usually chosen uniformly at random among the ones satisfying the condition at step 3. However this is irrelevant in the analysis below.

Exercise 11.10 Consider a code from the (l, k) regular LDPC ensemble (with $l \geq 3$). Assume that the received message differs from the transmitted one only in one position. Show that the bit-flipping algorithm always corrects such an error.

Exercise 11.11 Assume now that the channel has introduced two errors. Draw the factor graph of a regular (l, k) code for which the bit-flipping algorithm is unable to recover such an error event. What can you say of the probability of this type of graphs in the ensemble?

In order to monitor the bit-flipping algorithm, it is useful to introduce the 'energy':

$$E(t) \equiv \text{ Number of parity check equations not satisfied by } \underline{x}(t)\,. \qquad (11.40)$$

This is a non-negative integer, and if $E(t) = 0$ the algorithm is halted and its output is $\underline{x}(t)$. Furthermore $E(t)$ cannot be larger than the number of parity checks M and decreases (by at least one) at each cycle. Therefore, the algorithm complexity is $O(N)$ (this is a commonly regarded as the ultimate goal for many communication problems).

It remains to be seen if the output of the bit-flipping algorithm is related to the transmitted codeword. In Fig. 11.7 we present the results of a numerical experiment. We considered the $(5, 10)$ regular ensemble and generated about 1000 random code and channel realizations for each value of the noise in some mesh. Then we applied the above algorithm and plotted the fraction of successfully decoded blocks, as well as the residual energy $E_* = E(t_*)$, where t_* is the total number of iterations of the algorithm. The data suggests that bit-flipping is able to overcome a finite noise level: it recovers the original message with high probability when less than about 2.5% of the bits are corrupted by the channel. Furthermore, the curves for $\mathrm{P}_{\mathrm{B}}^{\mathrm{bf}}$ under bit-flipping decoding become steeper and steeper as the system size is increased. It is natural to conjecture that asymptotically, a phase transition takes place at a well defined noise level p_{bf}: $\mathrm{P}_{\mathrm{B}}^{\mathrm{bf}} \to 0$ for $p < p_{\mathrm{bf}}$ and $\mathrm{P}_{\mathrm{B}}^{\mathrm{bf}} \to 1$ for $p > p_{\mathrm{bf}}$. Numerically $p_{\mathrm{bf}} = 0.025 \pm 0.005$.

This threshold can be compared with the one for MAP decoding: The results in Table 11.1 imply $0.108188 \le p_{\mathrm{MAP}} \le 0.109161$ for the $(5, 10)$ ensemble. Bit-flipping is significantly sub-optimal, but is still surprisingly good, given the extreme simplicity of the algorithm.

Can we provide any *guarantee* on the performances of the bit-flipping decoder? One possible approach consists in using the expansion properties of the underlying factor graph. Consider a graph from the (l, k) ensemble. We say that it is an (ε, δ)-**expander** if, for any set U of variable nodes such that $|U| \le N\varepsilon$, the set $|D|$ of neighboring check nodes has size $|D| \ge \delta|U|$. Roughly speaking, if the factor graph is an expander with a large **expansion constant** δ, any small set of corrupted bits induces a large number of unsatisfied parity checks. The bit-flipping algorithm can exploit these checks to successfully correct the errors.

It turns out that random graphs are very good expanders. This can be understood as follows. Consider a fixed subset U. As long as U is small, the subgraph induced by U and the neighboring factor nodes D is a tree with high probability. If this is the case, elementary counting shows that $|D| = (l - 1)|U| + 1$. This would suggest that one can achieve an expansion factor (close to) $l - 1$, for small enough ε. Of course this argument has several flaws. First of all, the subgraph induced by U is a tree only if U has sub-linear size, but we are interested in all subsets U with $|U| \le \varepsilon N$ for some fixed N. Then, while most of the small subsets U are trees, we need to be sure that *all* subsets expand well. Nevertheless, one can prove that the heuristic expansion factor is essentially correct:

Proposition 11.5 *Consider a random factor graph $\mathcal{F}$ from the (l, k) ensemble. Then, for any $\delta < l - 1$, there exists a constant $\varepsilon = \varepsilon(\delta; l, k) > 0$, such that $\mathcal{F}$ is a (ε, δ) expander with probability approaching 1 as $N \to \infty$.*

In particular, this implies that, for $l \ge 5$, a random (l, k) regular factor graph is, with high probability a $(\varepsilon, \frac{3}{4}\, l)$ expander. In fact, this is enough to assure that the code will perform well at low noise level:

Theorem 11.6 *Consider a regular (l, k) LDPC code $\mathfrak{C}$, and assume that the corresponding factor graph is an $(\varepsilon, \frac{3}{4}\, l)$ expander. Then, the bit-flipping algorithm is able to correct any pattern of less then $N\varepsilon/2$ errors produced by a binary symmetric channel. In particular $\mathrm{P_B}(\mathfrak{C}) \to 0$ for communication over a BSC(p) with $p < \varepsilon/2$.*

Proof: As usual, we assume the channel input to be the all-zeros codeword $\underline{0}$. We denote by $w = w(t)$ the weight of $\underline{x}(t)$ (the current configuration of the bit-flipping algorithm), and by $E = E(t)$ the number of unsatisfied parity checks, as in Eq. (11.40). Finally, we call F the number of *satisfied* parity checks among the ones which are neighbors of at least one corrupted bit in $\underline{x}(t)$ (a bit is 'corrupted' if it takes value 1).

Assume first that $0 < w(t) \leq N\varepsilon$ at some time t. Because of the expansion property of the factor graph, we have $E+F > \frac{3}{4}\, l\, w$. On the other hand, every unsatisfied parity check is the neighbor of at least one corrupted bit, and every satisfied check which is the neighbor of some corrupted bit must involve at least two of them. Therefore $E + 2F \leq l\, w$. Eliminating F from the above inequalities, we deduce that $E(t) > \frac{1}{2}\, l\, w(t)$. Let $E_i(t)$ be the number of unsatisfied checks involving bit x_i. Then:

$$\sum_{i:x_i(t)=1} E_i(t) \geq E(t) > \frac{1}{2}\, l\, w(t)\,. \tag{11.41}$$

Therefore, there must be at least one bit having more unsatisfied than satisfied neighbors, and the algorithm does not halt.

Let us now start the algorithm with $w(0) \leq N\varepsilon/2$. It must halt at some time t_*, either with $E(t_*) = w(t_*) = 0$ (and therefore decoding is successful), or with $w(t_*) \geq N\varepsilon$. In this second case, as the weight of $\underline{x}(t)$ changes by one at each step, we have $w(t_*) = N\varepsilon$. The above inequalities imply $E(t_*) > Nl\varepsilon/2$ and $E(0) \leq lw(0) \leq Nl\varepsilon/2$. This contradicts the fact that $E(t)$ is a strictly decreasing function of t. Therefore the algorithm, started with $w(0) \leq N\varepsilon/2$ ends up in the $w = 0$, $E = 0$ state. □

The approach based on expansion of the graph has the virtue of pointing out one important mechanism for the good performance of random LDPC codes, namely the local tree-like structure of the factor graph. It also provides explicit lower bounds on the critical noise level $p_{\rm bf}$ for bit-flipping. However, these bounds turn out to be quite pessimistic. For instance, in the case of the $(5, 10)$ ensemble, it has been proved that a typical factor graph is an $(\varepsilon, \frac{3}{4}l) = (\varepsilon, \frac{15}{4})$ expander for $\varepsilon < \varepsilon_* \approx 10^{-12}$. On the other hand, numerical simulations, cf. Fig. 11.7, show that the bit flipping algorithm performs well up to noise levels much larger than $\varepsilon_*/2$.

Notes

Modern (post-Cook Theorem) complexity theory was first applied to coding by (Berlekamp, McEliecee and van Tilborg, 1978) who showed that maximum likelihood decoding of linear codes is NP-hard.

LDPC codes were first introduced by Gallager in his Ph.D. thesis (Gallager, 1963; Gallager, 1962), which is indeed older than these complexity results. An excellent detailed account of modern developments is provided by (Richardson and Urbanke, 2008).

Gallager proposal did not receive enough consideration at the time. One possible explanation is the lack of computational power for simulating large codes in the sixties. The rediscovery of LDPC codes in the nineties (MacKay, 1999) was (at least in part) a consequence of the invention of Turbo codes by (Berrou and Glavieux, 1996). Both these classes of codes were soon recognized to be prototypes of a larger family: codes on sparse graphs.

The major technical advance after this rediscovery has been the introduction of irregular ensembles (Luby, Mitzenmacher, Shokrollahi, Spielman and Stemann, 1997; Luby, Mitzenmacher, Shokrollahi and Spielman, 1998). There exist no formal proof of the 'equivalence' (whatever this means) of the various possible definitions of LDPC ensembles in the large block-length limit. But as we will see in Ch. 15, the main property that enters in the analysis of LDPC ensembles is the local tree-like structure of the factor graph described in Sec. 9.5.1; and this property is rather robust with respect to a change of the ensemble.

Gallager (Gallager, 1963) was the first to compute the expected weight enumerator for regular ensembles, and to use it in order to bound the threshold for reliable communication. General ensembles were considered in (Litsyn and Shevelev, 2003; Burshtein and Miller, 2004; Di, Richardson and Urbanke, 2006). It turns out that the expected weight enumerator coincides with the typical (most likely) one to leading exponential order for regular ensembles (in statistical physics jargon: the annealed computation coincides with the quenched one). This is not the case for irregular ensembles, as pointed out in (Di, Montanari and Urbanke, 2004).

Proposition 11.2 is essentially known since (Gallager, 1963). The formulation quoted here is from (Méasson, Montanari and Urbanke, 2005*a*). This paper contains some examples of 'exotic' LDPC ensembles such that the maximum of the expected weight enumerator is at weight $w = N\omega_*$, with $\omega_* \neq 1/2$.

A proof of the upper bound 11.4 can be found in (Gallager, 1963). For some recent refinements, see (Burshtein, Krivelevich, Litsyn and Miller, 2002).

Bit-flipping algorithms played an important role in the revival of LDPC codes, especially following the work of Sipser and Spielman (Sipser and Spielman, 1996). These authors focused on explicit code construction based on expander graph. They also provide bounds on the expansion of random $\mathrm{LDPC}_N(l, k)$ codes. The lower bound on the expansion mentioned in Sec. 11.4 is taken from (Richardson and Urbanke, 2008).

12
Spin glasses

We have already encountered several examples of spin glasses in chapters 2 and **??**. Like most problems in equilibrium statistical physics, they can be formulated in the general framework of factor graphs. Spin glasses are disordered systems, whose magnetic properties are dominated by randomly placed impurities. The theory aims at describing the behavior of a typical sample of such materials. This motivates the definition and study of spin glass ensembles.

In this chapter we shall explore the glass phase of these models. It is useful to have a good understanding of glass phases as we shall see them appearing in various problems from optimization or coding theory. In general the occurence of a glass phase is described physically in terms of a dramatic slowdown in a dynamical relaxation process. Here we will focus instead on purely static characterizations of the glass phases, which can be applied to a broad class of problems. The focus of our presentation is on so-called 'mean field models', for at least two reasons: (*i*) A deep mathematical theory (still under developement) provides a precise understanding of their behavior; (*ii*) The ensembles of combinatorial optimization, and coding problems to be considered in the following fall naturally in this class. We shall discuss the two types of spin glass transitions that have been encountered such models.

In contrast to these 'soluble' cases, it must be stressed that very little is known (let alone proven) for realistic models of real spin glass materials. Even the existence of a spin glass phase is not established rigorously in this last case.

We first discuss in Sec. 12.1 how Ising models and their generalizations can be formulated in terms of factor graphs, and introduce several ensembles of these models. Frustration is a crucial feature of spin glasses; in Sec. 12.2 we discuss it in conjunction with gauge transformations. This section also explains how to derive some exact results with the sole use of gauge transformations. Sec. 12.3 describes the spin glass phase and the main approaches to its characterization. Finally, the phase diagram of a spin glass model with several glassy phases is traced in Sec. 12.4.

12.1 Spin glasses and factor graphs

12.1.1 Generalized Ising models

Let us recall the main ingredients of magnetic systems with interacting Ising spins. The variables are N Ising spins $\underline{\sigma} = \{\sigma_1, \ldots, \sigma_N\}$ taking values in $\{+1, -1\}$. These are jointly distributed according to Boltzmann's law for the energy function:

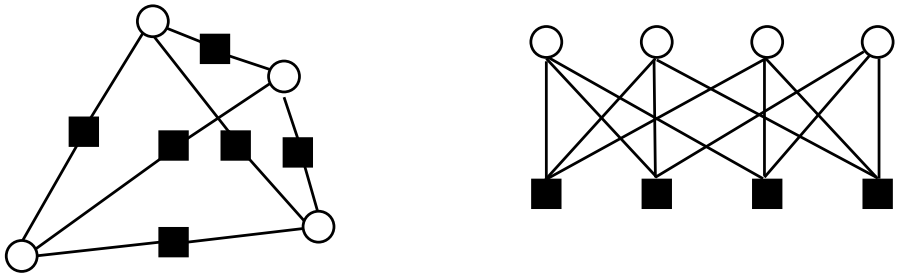

Fig. 12.1 Factor graph representation of the SK model with $N = 4$ (left), and the fully–connected 3-spin model with $N = 4$ (right). The squares denote the interactions between the spins.

$$E(\underline{\sigma}) = -\sum_{p=1}^{p_{\max}} \sum_{i_1<\cdots<i_p} J_{i_1\dots i_p}\sigma_{i_1}\cdots\sigma_{i_p} \ . \tag{12.1}$$

The index p gives the order of the interaction. One-body terms ($p = 1$) are also referred to as external field interactions, and will be sometimes written as $-B_i\sigma_i$. If $J_{i_1\dots i_p} \geq 0$, for any $i_1 \dots i_p$, and $p \geq 2$, the model is said to be a ferromagnet. If $J_{i_1\dots i_p} \leq 0$, it is an **antiferromagnet**. Finally, if both positive and negative couplings are present for $p \geq 2$, the model is a spin glass.

The energy function can be rewritten as $E(\underline{\sigma}) = \sum_a E_a(\underline{\sigma}_{\partial a})$, where $E_a(\underline{\sigma}_{\partial a}) \equiv -J_a\sigma_{i_1^a}\cdots\sigma_{i_{p_a}^a}$. Each interaction term a involves the spins contained in a subset $\underline{\sigma}_{\partial a} = \{\sigma_{i_1^a}, \dots, \sigma_{i_{p_a}^a}\}$, of size p_a. We then introduce a factor graph in which each interaction term is represented by a square vertex and each spin is represented by a circular vertex. Edges are drawn between the interaction vertex a and the variable vertex i whenever the spin σ_i appears in $\underline{\sigma}_{\partial a}$. We have already seen in Fig. 9.7 the factor graph of a 'usual' two-dimensional spin glass, where the energy contains terms with $p = 1$ and $p = 2$. Fig. 12.1 shows the factor graphs of some small samples of the SK model in zero magnetic field ($p = 2$ only) and the '3-spin model' in which terms with $p = 3$ appear in the energy function.

The energy function (12.1) can be straightforwardly interpreted as a model for a magnetic system. We used so far the language inherited from this application: the spins $\{\sigma_i\}$ are 'rotational' degrees of freedom associated to magnetic particle, their average is the magnetization etc. In this context, the most relevant interaction between distinct degrees of freedom is pairwise: $-J_{ij}\sigma_i\sigma_j$.

Higher order terms naturally arise in other applications, one of the simplest one being lattice particle systems. These are used to model the liquid-to-gas, liquid-to-solid, and similar phase transitions. One normally starts by considering some base graph $\mathcal{G}$ over N vertices, which is often taken to be a portion of $\mathbb{Z}^d$ (to model a real physical system the dimension of choice is of course $d = 3$). Each vertex in the graph can be either occupied by a particle, which we shall assume indistinguishable from the others, or empty. The particles are assumed indistinguishable from each other, and a configuration is characterized by occupation variables $n_i = \{0, 1\}$. The energy is

a function $E(\underline{n})$ of the occupancies $\underline{n} = \{n_1, \dots, n_N\}$, which takes into account local interaction among neighboring particles. Usually it can be rewritten in the form (12.1), using the mapping $\sigma_i = 1 - 2n_i$. We give a few examples in the exercises below.

Exercise 12.1 Consider an empty box which is free to exchange particles with a reservoir, and assume that particles do not interact with each other (except for the fact that they cannot superimpose). This can be modeled by taking $\mathcal{G}$ to be a cube of side L in $\mathbb{Z}^d$, and establishing that each particle in the system contributes by a constant amount $-\mu$ to the energy: $E(\underline{n}) = -\mu \sum_i n_i$. This is a model for what is usually called an **ideal gas**.

Compute the partition function. Rewrite the energy function in terms of spin variables and draw the corresponding factor graph.

Exercise 12.2 In the same problem, imagine that particles attract each other at short distance: whenever two neighboring vertices i and j are occupied, the system gains an energy $-\epsilon$. This is a model for the liquid-gas phase transition.

Write the corresponding energy function both in terms of occupancy variables $\{n_i\}$ and spin variables $\{\sigma_i\}$. Draw the corresponding factor graph. Based on the phase diagram of the Ising model, cf. Sec. 2.5, discuss the behavior of this particle system. What physical quantity corresponds to the magnetization of the Ising model?

Exercise 12.3 In some materials, molecules cannot be packed in a regular lattice at high density, and this may result in amorphous solid materials. In order to model this phenomenon, one can modify the energy function of the previous exercises as follows. Each time that a particle (i.e. an occupied vertex) is surrounded by more than k other particles in the neighboring vertices, a penalty $+\delta$ is added to the energy.

Write the corresponding energy function (both in terms of $\{n_i\}$ and $\{\sigma_i\}$) and draw the factor graph associated with it.

12.1.2 Spin glass ensembles

A sample (or an instance) of a spin glass is defined by:

- Its factor graph, which specifies the subsets of spins which interact;
- The value of the coupling constant $J_a \in \mathbb{R}$ for each function node in the factor graph.

An ensemble is defined by a probability distribution over the space of samples. In all cases which we shall consider, the couplings are assumed to be i.i.d. random variables, independent of the factor graph. The most studied cases are Gaussian J_a's, or J_a taking values $\{+1, -1\}$ with equal probability (in jargon this is called the $\pm J$ model). More generally, we shall denote by $\mathcal{P}(J)$ the pdf of J_a.

One can distinguish two large families of spin glass ensembles which have attracted the attention of physicists: 'realistic' and 'mean field' ones. While in the first case the focus is on modeling actual physical systems, mean field models have proved to

be analytically tractable, and revealed a rich mathematical structure. The relation between these two classes is a fascinating open problem that we will not try to address.

Physical spin glasses are mostly three-dimensional systems, but in some cases they can be two-dimensional. The main feature of realistic ensembles is that they retain this geometric structure: a position x in d dimensions can be associated with each spin. The interaction strength (the absolute value of the coupling J) decays rapidly with the distance between the positions of the associated spins. The Edwards-Anderson model is the most studied example of this family. The spins are located on the vertices of a d-dimensional hyper-cubic lattice. Neighboring spins interact, through two-body interactions (i.e. $p_{\max} = 2$ in Eq. (12.1)). The corresponding factor graph is not random, as can be seen on the two-dimensional example of Fig. 9.7. The only source of disorder are the random couplings J_{ij} distributed according to $\mathcal{P}(J)$. It is customary to add a uniform magnetic field B which is written as a $p = 1$ term with $J_i = B$. Very little is known about these models when $d \geq 2$, and most of our knowledge comes from numerical simulations. They suggest the existence of a glass phase when $d \geq 3$ but this is not proven yet.

There exists no general mathematical definition of mean field models. From a technical point of view, mean field models admit exact expressions for the asymptotic ($N \to \infty$) free-energy density, as the optimum of some sort of large deviation rate function. The distinctive feature allowing for a solution in this form is the lack of any finite-dimensional geometrical structure.

The p-spin glass model discussed in Sec. **??** (and in particular the $p = 2$ case, which is the SK model) is a mean field model. Also in this case the factor graph is non-random, and the disorder enters only in the random couplings. The factor graph is a regular bipartite graph. It contains $\binom{N}{p}$ function nodes, one for each p-uple of spins; for this reason it is called **fully connected**. Each function node has degree p, each variable node has degree $\binom{N-1}{p-1}$. Since the degree diverges with N, the coupling distribution $\mathcal{P}(J)$ must be scaled appropriately with N, cf. Eq. (**??**).

Fully connected models are among the best understood in the mean field family. They can be studied either via the replica method, as in Ch. **??**, or via the cavity method that we shall develop in the next chapters. Some of the predictions from these two heuristic approaches have been confirmed rigorously.

One unrealistic feature of fully connected models is that each spin interacts with a diverging number of other spins (the degree of a spin variable in the factor graph diverges in the thermodynamic limit). In order to eliminate this feature, one can study spin glass models on Erdös-Rényi random graphs with finite average degree. Spins are associated with vertices in the graph and $p = 2$ interactions (with couplings that are i.i.d. random variables drawn from $\mathcal{P}(J)$) are associated with edges in the graph. The generalization to p-spin interactions is immediate. The corresponding spin glass models will be named **diluted spin glasses (DSG)**. We define the ensemble $\mathsf{DSG}_N(p, M, \mathcal{P})$ as follows:

- Generate a factor graph from the $\mathbb{G}_N(p, M)$ ensemble (the graph has therefore M function nodes, all of degree p);
- For every function node a in the graph, connecting spins $i_1^a, \dots, i_p^a$, draw a random coupling $J_{i_1^a,\dots,i_p^a}$ from the distribution $\mathcal{P}(J)$, and introduce an energy term;

$$E_a(\underline{\sigma}_{\partial a}) = -J_{i_1^a,\dots,i_p^a}\sigma_{i_1^a}\cdots\sigma_{i_p^a} \ ; \tag{12.2}$$

- The final energy is $E(\underline{\sigma}) = \sum_{a=1}^{M} E_a(\underline{\sigma}_{\partial a})$.

The thermodynamic limit is taken by letting $N \to \infty$ at fixed $\alpha = M/N$.

As in the case of random graphs, one can introduce some variants of this definition. In the ensemble $\mathsf{DSG}(p,\alpha,\mathcal{P})$, the factor graph is drawn from $\mathbb{G}_N(p,\alpha)$: each p-uple of variable nodes is connected by a function node independently with probability $\alpha/\binom{N}{p}$. As we shall see, the ensembles $\mathsf{DSG}_N(p,M,\mathcal{P})$ and $\mathsf{DSG}_N(p,\alpha,\mathcal{P})$ have the same free-energy per spin in the thermodynamic limit, and many of their thermodynamic properties are identical. One basic reason of this phenomenon is that any finite neighborhood of a random site i has the same asymptotic distribution in the two ensembles.

Obviously, any ensemble of random graphs can be turned into an ensemble of spin glasses by the same procedure. Some of these ensembles have been considered in the literature. Mimicking the notation defined in Sect. 9.2, we shall introduce general diluted spin glasses with constrained degree profiles, to be denoted by $\mathsf{DSG}_N(\Lambda,P,\mathcal{P})$, as the ensemble derived from the random graphs in $\mathbb{D}_N(\Lambda,P)$.

Diluted spin glasses are a very interesting class of models, which are intimately related to sparse graph codes and to random satisfiability problems, among others. Our understanding of DSGs is intermediate between fully connected models and realistic ones. It is believed that both the replica and cavity methods should allow to compute exactly many thermodynamic properties for most of these models. However the number of these exact results is still rather small, and only a fraction of these have been proved rigorously.

12.2 Spin glasses: Constraints and frustration

Spin glasses at zero temperature can be seen as constraint satisfaction problems. Consider for instance a model with two-body interactions

$$E(\underline{\sigma}) = -\sum_{(i,j)\in\mathcal{E}} J_{ij}\sigma_i\sigma_j \ , \tag{12.3}$$

where the sum is over the edge set $\mathcal{E}$ of a graph $\mathcal{G}$ (the corresponding factor graph is obtained by associating a function node a to each edge $(ij) \in \mathcal{E}$). At zero temperature the Boltzmann distribution is concentrated on those configurations which minimize the energy. Each edge (i,j) induces therefore a constraint between the spins σ_i and σ_j: they should be aligned if $J_{ij} > 0$, or anti-aligned if $J_{ij} < 0$. If there exists a spin configuration which satisfies all the constraint, the ground state energy is $E_{\rm gs} = -\sum_{(i,j)\in\mathcal{E}} |J_{ij}|$ and the sample is said to be **unfrustrated** (see Ch. 2.6). Otherwise it is **frustrated**. In this case one defines a ground state as a spin configuration which violates the minimum possible number of constraints.

As shown in the Exercise below, there are several methods to check whether an energy function of the form (12.3) is frustrated.

Exercise 12.4 Define a 'plaquette' of the graph as a circuit $i_1, i_2, \dots, i_L, i_1$ such that no shortcut exists: $\forall r, s \in \{1, \dots, L\}$, the edge (i_r, i_s) is absent from the graph whenever $r \neq s \pm 1 \pmod L$. Show that a spin glass sample is unfrustrated if and only if the product of the couplings along every plaquette of the graph is positive.

Exercise 12.5 Consider a spin glass of the form (12.3), and define the Boolean variables $x_i = (1-\sigma_i)/2$. Show that the spin glass constraint satisfaction problem can be transformed into an instance of the 2-satisfiability problem. [Hint: Write the constraint $J_{ij}\sigma_i\sigma_j > 0$ in Boolean form using x_i and x_j.]

Since 2-SAT is in P, and because of the equivalence explained in the last exercise, one can check in polynomial time whether the energy function (12.3) is frustrated or not. This approach does not work when $p \geq 3$ because K-SAT is NP-complete for $K \geq 3$. However, as we shall see in Ch. **??**, checking whether a spin glass energy function is frustrated remains a polynomial problem for any p.

12.2.1 Gauge transformation

When a spin glass sample has some negative couplings but is unfrustrated, one is in fact dealing with a 'disguised ferromagnet'. By this we mean that, through a change of variables, the problem of computing the partition function for such a system can be reduced to the one of computing the partition function of a ferromagnet. Indeed, by assumption, there exists a ground state spin configuration $\sigma_i^* \in \{\pm 1\}$ such that $\forall (i,j) \in \mathcal{E}\ \ J_{ij}\sigma_i^*\sigma_j^* > 0$. Given a configuration $\underline{\sigma}$, define $\tau_i = \sigma_i\sigma_i^*$, and notice that $\tau_i \in \{+1, -1\}$. Then the energy of the configuration is $E(\underline{\sigma}) = E_*(\underline{\tau}) \equiv -\sum_{(i,j)\in\mathcal{E}} |J_{ij}|\tau_i\tau_j$. Obviously the partition function for the system with energy function $E_*(\,\cdot\,)$ (which is a ferromagnet since $|J_{ij}| > 0$) is the same as for the original system.

This change of variables is an example of a **gauge transformation**. In general, such a transformation amounts to changing all spins and simultaneously all couplings according to:

$$\sigma_i \mapsto \sigma_i^{(\underline{s})} = \sigma_i s_i \quad , \quad J_{ij} \mapsto J_{ij}^{(\underline{s})} = J_{ij} s_i s_j \ , \tag{12.4}$$

where $\underline{s} = \{s_1, \dots, s_N\}$ is an arbitrary configuration in $\{-1, 1\}^N$. If we regard the partition function as a function of the coupling constants $\underline{J} = \{J_{ij} : (ij) \in \mathcal{E}\}$:

$$Z[\underline{J}] = \sum_{\{\sigma_i\}} \exp\left(\beta \sum_{(ij)\in\mathcal{E}} J_{ij}\sigma_i\sigma_j\right) , \tag{12.5}$$

then we have

$$Z[\underline{J}] = Z[\underline{J}^{(\underline{s})}] \, . \tag{12.6}$$

The system with coupling constants $\underline{J}^{(\underline{s})}$ is sometimes called the 'gauge transformed system'.

Exercise 12.6 Consider adding a uniform magnetic field (i.e. a linear term of the form $-B\sum_i \sigma_i$) to the energy function (12.3), and apply a generic gauge transformation to such a system. How must the uniform magnetic field be changed in order to keep the partition function unchanged? Is the new magnetic field term still uniform?

Exercise 12.7 Generalize the above discussion of frustration and gauge transformations to the $\pm J$ 3-spin glass (i.e. a model of the type (12.1) involving only terms with $p = 3$).

12.2.2 The Nishimori temperature...

In many spin glass ensembles, there exists a special temperature (called the **Nishimori temperature**) at which some thermodynamic quantities, such as the internal energy, can be computed exactly. This nice property is particularly useful in the study of inference problems (a particular instance being symbol MAP decoding of error correcting codes), since the Nishimori temperature naturally arises in these context. There are in fact two ways of deriving it: either as an application of gauge transformations (this is how it was discovered in physics), or by mapping the system onto an inference problem.

Let us begin by taking the first point of view. Consider, for the sake of simplicity, the model (12.3). The underlying graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ can be arbitrary, but we assume that the couplings J_{ij} on all the edges $(ij) \in \mathcal{E}$ are i.i.d. random variables taking values $J_{ij} = +1$ with probability $1 - p$ and $J_{ij} = -1$ with probability p. We denote by $\mathbb{E}$ the expectation with respect to this distribution.

The Nishimori temperature for this system is given by $T_{\rm N} = 1/\beta_{\rm N}$, where $\beta_{\rm N} = \frac{1}{2}\log\frac{(1-p)}{p}$. It is chosen in such a way that the coupling constant distribution $\mathcal{P}(J)$ satisfies the condition:

$$\mathcal{P}(J) = e^{-2\beta_{\rm N} J}\, \mathcal{P}(-J)\,. \tag{12.7}$$

An equivalent way of stating the same condition consists in writing

$$\mathcal{P}(J) = \frac{e^{\beta_{\rm N} J}}{2\cosh(\beta_{\rm N} J)}\; \mathcal{Q}(|J|)\,. \tag{12.8}$$

where $\mathcal{Q}(|J|)$ denotes the distribution of the absolute values of the couplings (in the present example, this is a Dirac's delta on $|J| = 1$).

Let us now turn to the computation of the average, over the distribution of couplings, of the internal energy[1] $U \equiv \mathbb{E}\langle E(\underline{\sigma})\rangle$. More explicitly

$$U = \mathbb{E}\left\{ \frac{1}{Z[\underline{J}]} \sum_{\underline{\sigma}} \Big(-\sum_{(kl)} J_{kl}\sigma_k\sigma_l \Big)\, e^{\beta \sum_{(ij)} J_{ij}\sigma_i\sigma_j} \right\}, \tag{12.9}$$

[1]The same symbol U was used in Ch. 2 to denote the internal energy $\langle E(\underline{\sigma})\rangle$ (instead of its average). There should be no confusion with the present use.

In general, it is very difficult to compute U. It turns out that at the Nishimori temperature, the gauge invariance allows for an easy computation. The average internal energy U can be expressed as $U = \mathbb{E}\{Z_U[\underline{J}]/Z[\underline{J}]\}$, where $Z_U[\underline{J}] = -\sum_{\underline{\sigma}} \sum_{(kl)} J_{kl}\sigma_k\sigma_l \prod_{(ij)} e^{\beta_{\rm N} J_{ij}\sigma_i\sigma_j}$.

Let $\underline{s} \in \{-1,1\}^N$. By an obvious generalization of (12.6), we have $Z_U[\underline{J}^{(\underline{s})}] = Z_U[\underline{J}]$, and therefore

$$U = 2^{-N} \sum_{\underline{s}} \mathbb{E}\{Z_U[\underline{J}^{(\underline{s})}]/Z[\underline{J}^{(\underline{s})}]\} \,. \tag{12.10}$$

If the coupling constants J_{ij} are i.i.d. with distribution (12.8), then the gauge transformed constants $J'_{ij} = J_{ij}^{(\underline{s})}$ are equally independent but with distribution

$$\mathcal{P}_{\underline{s}}(J_{ij}) = \frac{e^{\beta_{\rm N} J_{ij} s_i s_j}}{2\cosh\beta_{\rm N}} \,. \tag{12.11}$$

Equation (12.10) can therefore be written as $U = 2^{-N} \sum_{\underline{s}} \mathbb{E}_{\underline{s}}\{Z_U[\underline{J}]/Z[\underline{J}]\}$, where $\mathbb{E}_{\underline{s}}$ denotes expectation with respect to the modified measure $\mathcal{P}_{\underline{s}}(J_{ij})$. Using Eq. (12.11), and denoting by $\mathbb{E}_0$ the expectation with respect to the uniform measure over $J_{ij} \in \{\pm 1\}$, we get

$$U = 2^{-N} \sum_{\underline{s}} \mathbb{E}_0 \left\{ \prod_{(ij)} \frac{e^{\beta_{\rm N} J_{ij} s_i s_j}}{\cosh\beta_{\rm N}} \, \frac{Z_U[\underline{J}]}{Z[\underline{J}]} \right\} = \tag{12.12}$$

$$= 2^{-N} (\cosh\beta_{\rm N})^{-|\mathcal{E}|} \mathbb{E}_0 \left\{ \sum_{\underline{s}} e^{\beta_{\rm N} \sum_{(ij)} J_{ij} s_i s_j} \, \frac{Z_U[\underline{J}]}{Z[\underline{J}]} \right\} = \tag{12.13}$$

$$= 2^{-N} (\cosh\beta_{\rm N})^{-|\mathcal{E}|} \mathbb{E}_0 \left\{ Z_U[\underline{J}] \right\} \,. \tag{12.14}$$

It is easy to compute $\mathbb{E}_0 Z_U[\underline{J}] = -2^N (\cosh\beta_{\rm N})^{|\mathcal{E}|-1} \, \sinh\beta_{\rm N}$. This implies our final result for the average energy at the Nishimori temperature:

$$U = -|\mathcal{E}| \tanh(\beta_{\rm N}) \,. \tag{12.15}$$

Notice that this simple result holds for any choice of the underlying graph. Furthermore, it is easy to generalize it to other choices of the coupling distribution satisfying Eq. (12.8) and to models with multi-spin interactions of the form (12.1). An even wider generalization is treated below.

12.2.3 . . . and its relation with probability

The calculation of the internal energy in the previous Section is straightforward but somehow mysterious. It is hard to grasp what is the fundamental reason that make things simpler at the Nishimori temperature. Here we discuss a more general derivation, in a slightly more abstract setting, which is related to the connection with inference mentioned above.

Consider the following process. A configuration $\underline{\sigma} \in \{\pm 1\}$ is chosen uniformly at random, we call $\mathbb{P}_0(\underline{\sigma})$ the corresponding distribution. Next a set of coupling constants $\underline{J} = \{J_a\}$ is chosen according to the conditional distribution

$$\mathbb{P}(\underline{J}|\underline{\sigma}) = e^{-\beta E_{\underline{J}}(\underline{\sigma})}\, \mathbb{Q}_0(\underline{J})\,. \tag{12.16}$$

Here $E_{\underline{J}}(\underline{\sigma})$ is an energy function with coupling constants $\underline{J}$, and $\mathbb{Q}_0(\underline{J})$ is some reference measure (that can be chosen in such a way that the resulting $\mathbb{P}(\underline{J}|\underline{\sigma})$ is normalized). This can be interpreted as a communication process. The information source produces the message $\underline{\sigma}$ uniformly at random, and the receiver observes the couplings $\underline{J}$.

The joint distribution of $\underline{J}$ and $\underline{\sigma}$ is $\mathbb{P}(\underline{J}, \underline{\sigma}) = e^{-\beta E_{\underline{J}}(\underline{\sigma})}\, \mathbb{Q}_0(\underline{J})\mathbb{P}_0(\underline{\sigma})$. We shall denote expectation with respect to this joint distribution by Av in order to distinguish it from the thermal average (the one over the Boltzmann measure, denoted by $\langle\, .\, \rangle$) and from the quenched average over the couplings, denoted by $\mathbb{E}$.

We assume that this process enjoys a gauge symmetry: this assumption defines the Nishimori temperature. By this we mean that, given $\underline{s} \in \{\pm 1\}^N$, there exists an invertible mapping $\underline{J} \to \underline{J}^{(\underline{s})}$ such that $\mathbb{Q}_0(\underline{J}^{(\underline{s})}) = \mathbb{Q}_0(\underline{J})$ and $E_{\underline{J}^{(\underline{s})}}(\underline{\sigma}^{(\underline{s})}) = E_{\underline{J}}(\underline{\sigma})$. Then it is clear that the joint probability distribution of the coupling and the spins, and the conditional one, enjoy the same symmetry

$$\mathbb{P}(\underline{\sigma}^{(\underline{s})}, \underline{J}^{(\underline{s})}) = \mathbb{P}(\underline{\sigma}, \underline{J})\,, \qquad \mathbb{P}(\underline{J}^{(\underline{s})}|\underline{\sigma}^{(\underline{s})}) = \mathbb{P}(\underline{J}|\underline{\sigma})\,. \tag{12.17}$$

Let us introduce the quantity

$$\mathcal{U}(\underline{J}) = \mathrm{Av}(E_{\underline{J}}(\underline{\sigma})|\underline{J}) = \sum_{\underline{\sigma}} \mathbb{P}(\underline{\sigma}|\underline{J}) E_{\underline{J}}(\underline{\sigma})\,. \tag{12.18}$$

and denote by $U(\underline{\sigma}_0) = \sum_{\underline{J}} \mathbb{P}(\underline{J}|\underline{\sigma}_0)\mathcal{U}(\underline{J})$. This is nothing but the average internal energy for a disordered system with energy function $E_{\underline{J}}(\underline{\sigma})$ and coupling distribution $\mathbb{P}(\underline{J}|\underline{\sigma}_0)$. For instance, if we take $\underline{\sigma}_0$ as the 'all-plus' configuration, $\mathbb{Q}_0(\underline{J})$ proportional to the uniform measure over $\{\pm 1\}^{\mathcal{E}}$, and $E_{\underline{J}}(\underline{\sigma})$ as given by Eq. (12.3), then $U(\underline{\sigma}_0)$ is exactly the quantity U that we computed in the previous Section.

Gauge invariance implies that $\mathcal{U}(\underline{J}) = \mathcal{U}(\underline{J}^{(\underline{s})})$ for any $\underline{s}$, and $U(\underline{\sigma}_0)$ does not depend upon $\underline{\sigma}_0$. We can therefore compute $U = U(\underline{\sigma}_0)$ by averaging over $\underline{\sigma}_0$. We obtain

$$\begin{aligned} U &= \sum_{\underline{\sigma}_0} \mathbb{P}_0(\underline{\sigma}_0) \sum_{\underline{J}} \mathbb{P}(\underline{J}|\underline{\sigma}_0) \sum_{\underline{\sigma}} \mathbb{P}(\underline{\sigma}|\underline{J}) E_{\underline{J}}(\underline{\sigma}) \\ &= \sum_{\underline{\sigma}, \underline{J}} \mathbb{P}(\underline{\sigma}, \underline{J}) E_{\underline{J}}(\underline{\sigma}) = \sum_{\underline{J}} \mathbb{P}(\underline{J}|\underline{\sigma}_0) E_{\underline{J}}(\underline{\sigma})\,, \end{aligned} \tag{12.19}$$

where we used gauge invariance, once more, in the last step. The final expression is generally easy to evaluate since the coublings J_a are generically independent under $\mathbb{P}(\underline{J}|\underline{\sigma}_0)$ In particular, it is straightforward to recover Eq. (12.15) for the case treated in the last Section.

Exercise 12.8 Consider a spin glass model on an arbitrary graph, with energy given by (12.3), and i.i.d. random couplings on the edges, drawn from the distribution $\mathcal{P}(J) = \mathcal{P}_0(|J|)e^{aJ}$. Show that the Nishimori inverse temperature is $\beta_N = a$, and that the internal energy at this point is given by: $U = -|\mathcal{E}| \sum_J \mathcal{P}_0(|J|)\, J\, \sinh(\beta_N J)$. In the case where $\mathcal{P}$ is a Gaussian distribution of mean J_0, show that $U = -|\mathcal{E}|J_0$.

12.3 What is a glass phase?

12.3.1 Spontaneous local magnetizations

In physics, a 'glass' is defined through its dynamical properties. For classical spin models such as the ones we are considering here, one can define several types of physically meaningful dynamics. For definiteness we use the single spin flip Glauber dynamics defined in Sec. **??**. The main features of our discussion should remain unchanged as far as we keep to local dynamics (i.e. a bounded number of spins is flipped at each step), which obey detailed balance.

Consider a system at equilibrium at time 0 (i.e., assume $\underline{\sigma}(0)$ to be distributed according to the Boltzmann distribution) and denote by $\langle\,\cdot\,\rangle_{\underline{\sigma}(0)}$ the expectation with respect to Glauber dynamics *conditional* to the initial configuration. Within a 'solid' [2] phase, spins are correlated with their initial value on long time scales:

$$\lim_{t\to\infty}\lim_{N\to\infty}\langle\sigma_i(t)\rangle_{\underline{\sigma}(0)} \equiv m_{i,\underline{\sigma}(0)} \neq \langle\sigma_i\rangle\,. \tag{12.20}$$

In other words, on arbitrary long but finite (in the system size) time scales, the system converges to a 'quasi-equilibrium' state, which we shall call for brevity 'quasi-state', with local magnetizations $m_{i,\underline{\sigma}(0)}$ depending on the initial condition.

The condition (12.20) is for instance satisfied by a $d \geq 2$ Ising ferromagnet in zero external field, at temperatures below the ferromagnetic phase transition. In this case we have either $m_{i,\underline{\sigma}(0)} = M(\beta)$, or $m_{i,\underline{\sigma}(0)} = -M(\beta)$ depending on the initial condition, where $M(\beta)$ is the spontaneous magnetization of the system. There are two quasi-states, invariant by translation and related by a simple symmetry transformation. If the different quasi-states are not periodic, nor related by any symmetry, one may speak of a glass phase.

It is also very important to characterize the glass phase at the level of equilibrium statistical mechanics, without introducing a specific dynamics. For the case of ferromagnets we have already seen the solution of this problem in Ch. 2. Let $\langle\,.\,\rangle_B$ denote expectation with respect to the Boltzmann measure for the energy function (12.1), after a uniform magnetic field has been added. One then defines the two quasi-states by:

$$m_{i,\pm} \equiv \lim_{B\to 0\pm}\lim_{N\to\infty}\langle\sigma_i\rangle_B\,. \tag{12.21}$$

A natural generalization to glasses consists in adding a small magnetic field which is not uniform. Let us add to the energy function (12.1) a term of the form $-\epsilon\sum_i s_i\sigma_i$

[2] The name comes from the fact that in a solid the preferred position of the atoms are time independent, for instance in a crystal they are the vertices of a periodic lattice

where $\underline{s} \in \{\pm 1\}^N$ is an arbitrary configuration. Denote by $\langle \cdot \rangle_{\epsilon,\underline{s}}$ the expectation with respect to the corresponding Boltzmann distribution and let

$$m_{i,\underline{s}} \equiv \lim_{\epsilon \to 0\pm} \lim_{N\to\infty} \langle \sigma_i \rangle_{\epsilon,\underline{s}} \, . \tag{12.22}$$

The **Edwards-Anderson order parameter**, defined as

$$q_{\rm EA} \equiv \lim_{\epsilon \to 0\pm} \lim_{N\to\infty} \frac{1}{N} \sum_i \langle \sigma_i \rangle^2_{\epsilon,\underline{s}} \, , \tag{12.23}$$

where $\underline{s}$ is an equilibrium configuration sampled from Boltzmann's distribution, then signals the onset of the spin glass phase.

The careful reader will notice that the Eq. (12.20) is not really completely defined: How should we take the $N \to \infty$ limit? Do the limits exist, how does the result depend on $\underline{s}$? These are subtle questions. They underly the problem of defining properly the pure states (extremal Gibbs states) in disordered systems. We will come back to these issues in Chapter **??**.

An extremely fruitful idea is instead to study glassy phases by comparing several equilibrated (i.e. drawn from the Boltzmann distribution) configurations of the system: one can then use one configuration as defining the direction of the polarizing field, as we just did for the Edwards-Anderson order parameter. Remarkably, this idea underlies the formal manipulations within the replica method.

We shall explain below in greater detail two distinct criteria, based on this idea, which can be used to define a glass phase. Before this, let us discuss a criterion of stability of the high temperature phase.

12.3.2 Spin glass susceptibility

Take a spin glass sample, with energy (12.1), and add to it a local magnetic field on site i, B_i. The magnetic susceptibility of spin j with respect to the field B_i is defined as the rate of change of $m_j = \langle \sigma_j \rangle_{B_i}$ with respect to B_i:

$$\chi_{ji} \equiv \left. \frac{{\rm d}m_j}{{\rm d}B_i} \right|_{B_i=0} = \beta(\langle \sigma_i \sigma_j \rangle - \langle \sigma_i \rangle \langle \sigma_j \rangle) \, , \tag{12.24}$$

where we used the fluctuation dissipation relation (2.44).

The uniform (ferromagnetic) susceptibility defined in Sec. 2.5.1 gives the rate of change of the average magnetization with respect to an infinitesimal global uniform field: $\chi = \frac{1}{N} \sum_{i,j} \chi_{ji}$. Consider a ferromagnetic Ising model as introduced in Sec. 2.5. Within the ferromagnetic phase (i.e. at zero external field and below the critical temperature) χ diverges with the system size N. One way to understand this divergence is the following. If we denote by $m(B)$ the infinite volume magnetization in a magnetic field B, and by $M(\beta)$ the spontaneous magnetization, we have

$$\chi = \lim_{B\to 0} \frac{1}{2B}[m(B) - m(-B)] = \lim_{B\to 0+} M(\beta)/B = \infty \, , \tag{12.25}$$

within the ferromagnetic phase.

The above argument relates the susceptibility divergence with the existence of two distinct pure states of the system ('plus' and 'minus'). What is the appropriate susceptibility to detect a spin glass ordering? Following our previous discussion, we should consider the addition of a small non-uniform field $B_i = s_i\epsilon$. The local magnetizations are given by

$$\langle\sigma_i\rangle_{\epsilon,\underline{s}} = \langle\sigma_i\rangle_0 + \epsilon\sum_j \chi_{ij}s_j + O(\epsilon^2)\,. \tag{12.26}$$

As suggested by Eq. (12.25) we compare the local magnetization obtained by perturbing the system in two different directions $\underline{s}$ and $\underline{s}'$

$$\langle\sigma_i\rangle_{\epsilon,\underline{s}} - \langle\sigma_i\rangle_{\epsilon,\underline{s}'} = \epsilon\sum_j \chi_{ij}(s_j - s'_j) + O(\epsilon^2)\,. \tag{12.27}$$

How should we choose $\underline{s}$ and $\underline{s}'$? A simple choice takes them independent and uniformly random in $\{\pm 1\}^N$; let us denote by $\mathbb{E}_s$ the expectation with respect to this distribution. The above difference becomes therefore a random variable with zero mean. Its second moment allows to define the **spin glass susceptibility** (sometimes called **non-linear susceptibility**):

$$\chi_{\rm SG} \equiv \lim_{\epsilon\to 0}\frac{1}{2N\epsilon^2}\sum_i \mathbb{E}_s\left(\langle\sigma_i\rangle_{\epsilon,\underline{s}} - \langle\sigma_i\rangle_{\epsilon,\underline{s}'}\right)^2 \tag{12.28}$$

This is somehow the equivalent of Eq. (12.25) for the spin glass case. Using Eq. (12.27) one gets the expression $\chi_{\rm SG} = \frac{1}{N}\sum_{ij}(\chi_{ij})^2$, which can also be written, using the fluctuation dissipation relation, as:

$$\chi_{\rm SG} = \frac{\beta^2}{N}\sum_{i,j}\left[\langle\sigma_i\sigma_j\rangle - \langle\sigma_i\rangle\langle\sigma_j\rangle\right]^2\,. \tag{12.29}$$

Usually, a necessary condition for the system to be in a paramagnetic, non-solid, phase is that $\chi_{\rm SG}$ remain finite when $N\to\infty$. We shall see below that this necessary condition of local stability is not always sufficient.

Exercise 12.9 Another natural choice would consist in choosing $\underline{s}$ and $\underline{s}'$ as independent configurations drawn from Boltzmann's distribution. Show that with such a choice one would get $\chi_{\rm SG} = (1/N)\sum_{i,j,k}\chi_{ij}\chi_{jk}\chi_{ki}$. This susceptibility has not been studied in the literature, but it is reasonable to expect that it will lead generically to the same criterion of stability as the usual one (12.29).

12.3.3 The overlap distribution function $P(q)$

One of the main indicators of a glass phase is the overlap distribution, which we defined in Sect. **??**. Given a general magnetic model of the type (12.1), one generates two independent configurations $\underline{\sigma}$ and $\underline{\sigma}'$ from the associated Boltzmann distribution and consider their overlap $q_{\underline{\sigma},\underline{\sigma}'} = N^{-1}\sum_i \sigma_i\sigma'_i$. The overlap distribution $P_N(q)$ is the

distribution of $q_{\underline{\sigma},\underline{\sigma}'}$ when the couplings and the underlying factor graph are taken randomly from their ensemble. Its infinite N limit is denoted by $P(q)$. Its moments are given by:

$$\int P_N(q) q^r \, \mathrm{d}q = \mathbb{E}\Big\{ \frac{1}{N^r} \sum_{i_1,\dots,i_r} \langle \sigma_{i_1} \dots \sigma_{i_r} \rangle^2 \Big\} \, . \tag{12.30}$$

In particular, the first moment $\int P_N(q)\, q \;\mathrm{d}q = N^{-1} \sum_i m_i^2$ is the expected overlap and the variance $\mathrm{Var}(q) \equiv \int P_N(q)\, q^2 \;\mathrm{d}q - \big[\int P_N(q)\, q \;\mathrm{d}q\big]^2$ is related to the spin glass susceptibility:

$$\mathrm{Var}(q) = \mathbb{E}\Big\{ \frac{1}{N^2} \sum_{i,j} \big[\langle \sigma_i \sigma_j \rangle - \langle \sigma_i \rangle \langle \sigma_j \rangle \big]^2 \Big\} = \frac{1}{N} \chi_{\mathrm{SG}} \, . \tag{12.31}$$

How is a glass phase detected through the behavior of the overlap distribution $P(q)$? We will discuss here two scenarios that appear to be remarkably universal within mean field models. In the next section we will see that the overlap distribution is in fact related to the idea, discussed in Section 12.3.1, of perturbing the system in order to explore its quasi-states.

Generically, at small β, a system of the type (12.1) is found in a 'paramagnetic' or 'liquid' phase. In this regime $P_N(q)$ concentrates as $N \to \infty$ on a single (deterministic) value $q(\beta)$: with high probability, two independent configurations $\underline{\sigma}$ and $\underline{\sigma}'$ have overlap close to $q(\beta)$. In fact, in such a phase, the spin glass χ_{SG} susceptibility is finite, and the variance of $P_N(q)$ vanishes therefore as $1/N$.

For β larger than a critical value β_{c}, the distribution $P(q)$ may acquire some structure, in the sense that several values of the overlap have non-zero probability in the $N \to \infty$ limit. The temperature $T_{\mathrm{c}} = 1/\beta_{\mathrm{c}}$ is called the **static (or equilibrium) glass transition temperature**. . For $\beta > \beta_{\mathrm{c}}$ the system is in an equilibrium glass phase.

How does $P(q)$ look like at $\beta > \beta_{\mathrm{c}}$? Generically, the transition falls into one of the following two categories, the names of which come from the corresponding replica symmetry breaking pattern found in the replica approach:

(*i*) **Continuous** ("Full replica symmetry breaking -FRSB") glass transition. In Fig. 12.2 we sketch the behavior of the thermodynamic limit of $P(q)$ in this case. The delta function present at $\beta < \beta_{\mathrm{c}}$ 'broadens' for $\beta > \beta_{\mathrm{c}}$, giving rise to a distribution with support in some interval $[q_0(\beta), q_1(\beta)]$. The width $q_1(\beta) - q_0(\beta)$ vanishes continuously when $\beta \downarrow \beta_{\mathrm{c}}$. Furthermore, the asymptotic distribution has a continuous density which is strictly positive in $]q_0(\beta), q_1(\beta)[$ and two discrete (delta) contributions at $q_0(\beta)$ and $q_1(\beta)$.
This type of transition has a 'precursor'. If we consider the $N \to \infty$ limit of the spin glass susceptibility, this diverges as $\beta \uparrow \beta_{\mathrm{c}}$. This phenomenon is quite important for identifying the critical temperature experimentally, numerically and analytically.

(*ii*) **Discontinuous** ("1RSB") glass transition. Again, $P(q)$ acquires a non trivial structure in the glass phase, but the scenario is different. When β increases above β_{c}, the δ-peak at $q(\beta)$, which had unit mass at $\beta \le \beta_c$, becomes a peak at $q_0(\beta)$, with a mass $1 - \mathtt{x}(\beta) < 1$. Simultaneously, a second δ-peak appears at a value of

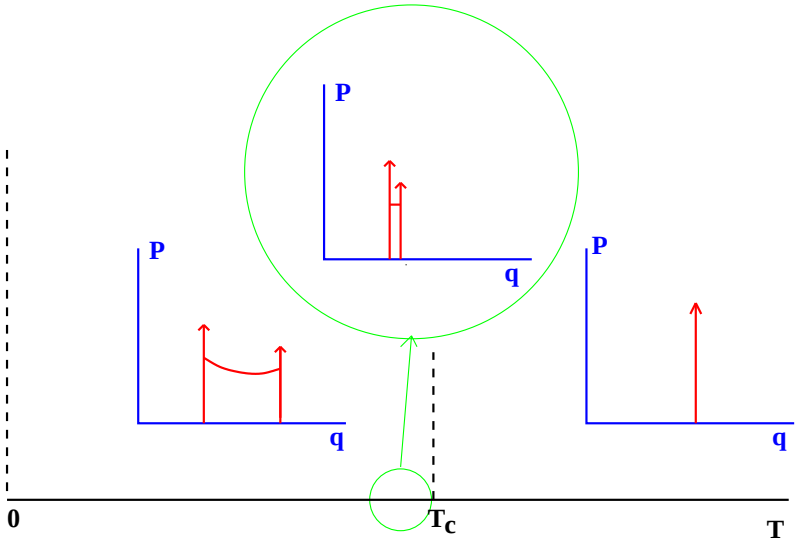

Fig. 12.2 Typical behavior of the order parameter $P(q)$ (asymptotic overlap distribution) at a continuous-FRSB glass transition. Vertical arrows denote Dirac's delta function.

the overlap $q_1(\beta) > q_0(\beta)$ with mass $\mathtt{x}(\beta)$. As $\beta \downarrow \beta_c$, $q_0(\beta) \to q(\beta_c)$ and $\mathtt{x}(\beta) \to 0$. Unlike in a continuous transition, the width $q_1(\beta) - q_0(\beta)$ does not vanish as $\beta \downarrow \beta_c$ and the open interval $]q_0(\beta), q_1(\beta)[$ has vanishing probability in the $N \to \infty$ limit. Furthermore, the thermodynamic limit of the spin glass susceptibility, χ_{SG} has a finite limit as $\beta \uparrow \beta_c$. This type of transition has no 'simple' precursor (but we shall describe below a more subtle indicator).

The two-peaks structure of $P(q)$ in a discontinuous transition has a particularly simple geometrical interpretation. When two configurations $\underline{\sigma}$ and $\underline{\sigma}'$ are chosen independently with the Boltzmann measure, their overlap is (with high probability) either approximately equal to q_0 or to q_1. In other words, their Hamming distance is either $N(1-q_1)/2$ or $N(1-q_0)/2$. This means that the Boltzmann measure $\mu(\underline{\sigma})$ is concentrated in some regions of the Hamming space $\{-1,1\}^N$, called **clusters**. With high probability, two independent random configurations in the same cluster have distance (close to) $N(1-q_1)/2$, and two configurations in distinct clusters have distance (close to) $N(1-q_0)/2$. In other words, while the overlap does not concentrate in probability when $\underline{\sigma}$ and $\underline{\sigma}'$ are drawn from the Boltzmann measure, it does when this measure is restricted to one cluster. In a more formal (but still imprecise) way, we might write

$$\mu(\underline{\sigma}) \approx \sum_\alpha w_\alpha \, \mu_\alpha(\underline{\sigma}) \,, \tag{12.32}$$

where the $p_\alpha(\,\cdot\,)$ are probability distributions concentrated onto a single cluster, and W_α are the weights attributed by the Boltzmann distribution to each cluster.

According to this interpretation, $\mathtt{x}(\beta) = \mathbb{E}\left\{\sum_\alpha w_\alpha^2\right\}$. Notice that, since $\mathtt{x}(\beta) > 0$ for $\beta > \beta_c$, the weights are sizeable only for a finite number of clusters (if there were

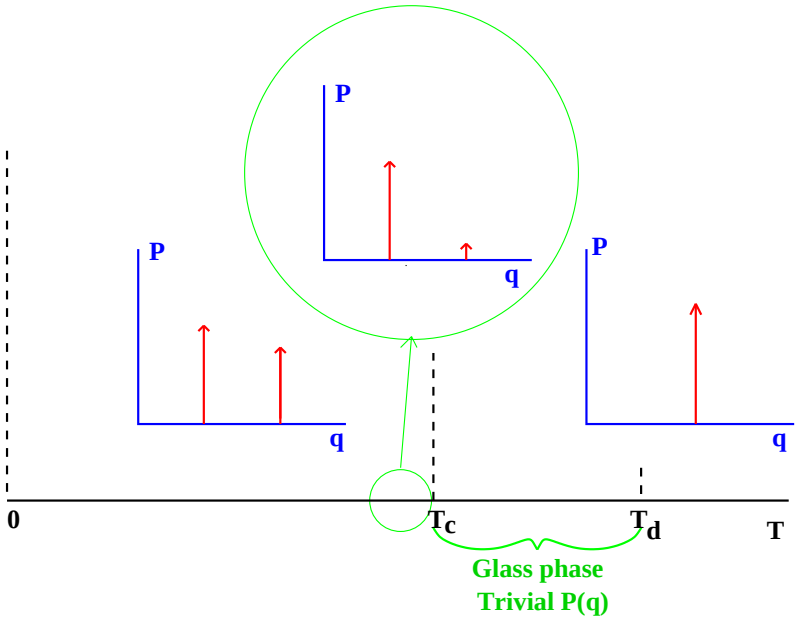

Fig. 12.3 Typical behavior of the order parameter $P(q)$ (overlap distribution) in a discontinuous-1RSB glass transition. Vertical arrows denote Dirac's delta function.

R clusters, all with the same weight $w_\alpha = 1/R$, one would have $\mathtt{x}(\beta) = 1/R$). This is what we found already in the REM, as well as in the replica solution of the completely connected p-spin model, cf. Sec. **??**.

Generically, clusters exist already in some region of temperatures above T_c, but the measure is not yet condensed on a finite number of them. The existence of clusters in this intermediate temperature region can be detected instead using the tools described below.

There is no clear criterion that allows to distinguish *a priori* between systems undergoing one or the other type of transition. The experience gained on models solved via the replica or cavity methods indicated that a continuous transition typically occurs in standard spin glasses with $p = 2$-body interactions, but also, for instance, in the vertex-cover problem. A discontinuous transition is instead found in structural glasses, generalized spin glasses with $p \geq 3$, random satisfiability and coloring. To complicate things, both types of transitions may occur in the same system at different temperatures (or varying some other parameter). This may lead to a rich phase diagram with several glass phases of different nature.

It is natural to wonder whether gauge transformations may give some information on $P(q)$. Unfortunately, it turns out that the Nishimori temperature never enters a spin glass phase: the overlap distribution at T_N is concentrated on a single value, as suggested by the next exercise.

Exercise 12.10 Using the gauge transformation of Sec. 12.2.1, show that, at the Nishimori temperature, the overlap distribution $P_N(q)$ is equal to the distribution of the magnetization per spin $m(\underline{\sigma}) \equiv N^{-1}\sum_i \sigma_i$. (In many spin glass models one expects that this distribution of magnetization per spin obeys a large deviation principle, and to concentrate on a single value as $N \to \infty$.)

12.3.4 The ϵ-coupling method

The overlap distribution is in fact related to the idea of quasi-states introduced in Sec. 12.3.1. Let us again consider a perturbation of the Boltzmann distribution defined by adding to the energy a magnetic field term $-\epsilon\sum_i s_i\sigma_i$, where $\underline{s} = (s_1, \dots, s_N)$ is a generic configuration. We introduce the ϵ-perturbed energy of a configuration $\underline{\sigma}$ as

$$E_{\epsilon,\underline{s}}(\underline{\sigma}) = E(\underline{\sigma}) - \epsilon\sum_{i=1}^{N} s_i\sigma_i \ . \tag{12.33}$$

Is is important to realize that both the original energy $E(\underline{\sigma})$ and the new term $-\epsilon\sum_i s_i\sigma_i$ are extensive, i.e. they grow proportionally to N as $N \to \infty$. Therefore in this limit the presence of the perturbation can be relevant. The ϵ-perturbed Boltzmann's measure is

$$\mu_{\epsilon,\underline{s}}(\underline{\sigma}) = \frac{1}{Z_{\epsilon,\underline{s}}}\, e^{-\beta E_{\epsilon,\underline{s}}(\underline{\sigma})} \ . \tag{12.34}$$

In order to quantify the effect of the perturbation, let us measure the expected distance between $\underline{\sigma}$ and $\underline{s}$

$$d(\underline{s},\epsilon) \equiv \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}(1 - s_i\langle\sigma_i\rangle_{\underline{s},\epsilon}) \tag{12.35}$$

(notice that $\sum_i(1 - s_i\sigma_i)/2$ is just the number of positions in which $\underline{\sigma}$ and $\underline{s}$ differ). For $\epsilon > 0$ the coupling between $\underline{\sigma}$ and $\underline{s}$ is attractive, for $\epsilon < 0$ it is repulsive. In fact it is easy to show that $d(\underline{s},\epsilon)$ is a decreasing function of ϵ.

In the **ϵ-coupling method**, $\underline{s}$ is taken as a random variable, drawn from the (unperturbed) Boltzmann distribution. The rationale for this choice is that in this way $\underline{s}$ will point in the directions corresponding to quasi-states. The average distance induced by the ϵ-perturbation is then obtained, after averaging over $\underline{s}$ and over the choice of sample:

$$d(\epsilon) \equiv \mathbb{E}\Big\{\sum_{\underline{s}}\frac{1}{Z}\, e^{-\beta E(\underline{s})}\, d(\underline{s},\epsilon)\Big\} . \tag{12.36}$$

There are two important differences between the ϵ-coupling method and the computation of the overlap distribution $P_N(q)$: (*i*) When computing $P_N(q)$, the two copies of the system are treated on equal footing: they are independent and distributed according to the Boltzmann law. In the ϵ-coupling method, one of the copies is distributed

according to Boltzmann's law, while the other follows a perturbed distribution depending on the first one. (*ii*) In the ϵ-coupling method the $N \to \infty$ limit is taken *at fixed* ϵ. Therefore, the sum in Eq. (12.36) can be dominated by values of the overlap $q(\underline{s}, \underline{\sigma})$ which would have been exponentially unlikely for the original (unperturbed) measure. When computing the $N \to \infty$ limit $P(q)$, such values of the overlap have a vanishing weight. The two approaches provide complementary informations.

Within a paramagnetic phase $d(\epsilon)$ remains a smooth function of ϵ in the neighborhood of $\epsilon = 0$, even after the $N \to \infty$ limit has been taken: perturbing the system does not have any dramatic effect. But in a glass phase $d(\epsilon)$ becomes singular: it develops a discontinuity at $\epsilon = 0$, that can be detected by defining

$$\Delta = \lim_{\epsilon \to 0+} \lim_{N \to \infty} d(\epsilon) - \lim_{\epsilon \to 0-} \lim_{N \to \infty} d(\epsilon)\,. \tag{12.37}$$

Notice that the limit $N \to \infty$ is taken first: for finite N there cannot be any discontinuity.

One expects Δ to be non-zero if and only if the system is in a 'solid' phase. In order to get an intuitive understanding, one can think of the process of adding a positive ϵ coupling and then letting it to 0, as of a physical process. The system is first forced in an energetically favorable configuration (given by $\underline{s}$). The forcing is then gradually removed and one checks whether any memory of the preparation is retained ($\Delta > 0$), or, vice-versa, the system 'liquefies' ($\Delta = 0$).

The advantage of the ϵ-coupling method with respect to the overlap distribution $P(q)$ is twofold:

- In some cases the dominant contribution to the Boltzmann measure comes from several distinct clusters, but a single one dominates over the others. More precisely, it may happen that the weights for sub-dominant clusters scale as $w_\alpha = \exp[-\Theta(N^\theta)]$, with $\theta \in]0, 1[$. In this case, $P(q)$ is a delta function and does not allow to distinguish from a purely paramagnetic phase. However, the ϵ-coupling method identifies the phase transition through a singularity of $d(\epsilon)$ at $\epsilon = 0$.
- One can use it to analyze a system undergoing a discontinuous transition, when it is in a glass phase but in the $T > T_{\rm c}$ regime. In this case, the existence of clusters cannot be detected from $P(q)$ because the Boltzmann measure is spread among an exponential number of them. This situation will be the object of the next section.

12.3.5 1RSB clusters and the potential method

The 1RSB equilibrium glass phase corresponds to a condensation of the measure on a small number of clusters of configurations. However, the most striking phenomenon is the appearance of clusters themselves. In the next chapters we will argue that this has important consequences on Monte Carlo dynamics as well as on other algorithmic approaches to these systems. It turns out that the Boltzmann measure splits into clusters at a distinct temperature $T_{\rm d} > T_{\rm c}$. In the region of temperatures $[T_{\rm c}, T_{\rm d}]$ we will say that the system is in a **clustered phase** or **dynamical glass phase** or **dynamical 1RSB phase**. The phase transition at $T_{\rm d}$ will be referred to as **clustering** or **dynamical (glass) transition**. In this regime, an exponential number of clusters

$\mathcal{N} \doteq e^{N\Sigma}$ carry a roughly equal weight. The rate of growth Σ is called **complexity**[3] or **configurational entropy**.

The thermodynamic limit of the overlap distribution, $P(q)$, does not show any signature of the clustered phase. In order to understand this point, it is useful to work out a toy example. Assume that the Boltzmann measure is entirely supported onto *exactly* $e^{N\Sigma}$ sets of configurations in $\{\pm 1\}^N$ (each set is a clusters), denoted by $\alpha = 1, \dots, e^{N\Sigma}$ and that the Boltzmann probability of each of these sets is $w = e^{-N\Sigma}$. Assume furthermore that, for any two configurations belonging to the same cluster $\underline{\sigma}, \underline{\sigma}' \in \alpha$, their overlap is $q_{\underline{\sigma},\underline{\sigma}'} = q_1$, while if they belong to different clusters $\underline{\sigma} \in \alpha$, $\underline{\sigma}' \in \alpha'$, $\alpha \neq \alpha'$ their overlap is $q_{\underline{\sigma},\underline{\sigma}'} = q_0 < q_1$. Although it might be actually difficult to construct such a measure, we shall neglect this for a moment, and compute the overlap distribution. The probability that two independent configurations fall in the same cluster is $e^{N\Sigma} w^2 = e^{-N\Sigma}$. Therefore, we have

$$P_N(q) = (1 - e^{-N\Sigma})\, \delta(q - q_0) + e^{-N\Sigma}\, \delta(q - q_1)\,, \tag{12.38}$$

which converges to $\delta(q - q_0)$ as $N \to \infty$: $P(q)$ has a single delta function, as in the paramagnetic phase.

A first signature of the clustered phase is provided by the ϵ-coupling method described in the previous Section. The reason is very clear if we look at Eq. (12.33): the epsilon coupling 'tilts' the Boltzmann distribution in such a way that unlikely values of the overlap acquire a strictly positive probability. It is easy to compute the thermodynamic limit $d_*(\epsilon) \equiv \lim_{N\to\infty} d(\epsilon)$. We get

$$d_*(\epsilon) = \begin{cases} (1 - q_0)/2 & \text{for } \epsilon < \epsilon_{\rm c}, \\ (1 - q_1)/2 & \text{for } \epsilon > \epsilon_{\rm c}, \end{cases} \tag{12.39}$$

where $\epsilon_{\rm c} = \Sigma/\beta(q_1 - q_0)$. As $T \downarrow T_{\rm c}$, clusters becomes less and less numerous and $\Sigma \to 0$. Correspondingly, $\epsilon_{\rm c} \downarrow 0$ as the equilibrium glass transition is approached.

The picture provided by this toy example is essentially correct, with the caveats that the properties of clusters will hold only within some accuracy and with high probability. Nevertheless, one expects $d_*(\epsilon)$ to have a discontinuity at some $\epsilon_{\rm c} > 0$ for all temperatures in an interval $]T_{\rm c}, T'_{\rm d}]$. Furthermore $\epsilon_{\rm c} \downarrow 0$ as $T \downarrow T_{\rm c}$.

In general, the temperature $T'_{\rm d}$ computed through the ϵ-coupling method does not coincide with the clustering transition. The reason is easily understood. As illustrated by the above example, we are estimating the exponentially small probability $\mathbb{P}(q|\underline{s}, \underline{J})$ that an equilibrated configuration $\underline{\sigma}$ has overlap q with the reference configuration $\underline{s}$, in a sample $\underline{J}$. In order to do this we compute the distance $d(\epsilon)$ in a problem with a tilted measure. As we have seen already several times since Ch. 5, exponentially small (or large) quantities, usually do not concentrate in probability, and $d(\epsilon)$ may be dominated by exponentially rare samples. We also learnt the cure for this problem: take logarithms! We therefore define[4] the **glass potential**

[3]This use of the term 'complexity', which is customary in statistical physics, should not be confused with its use in theoretical computer science.

[4]One should introduce a resolution, so that the overlap is actually constrained in some window around q. The width of this window can be let to 0 *after* $N \to \infty$.

$$V(q) = -\lim_{N\to\infty} \frac{1}{N\beta} \mathbb{E}_{\underline{s},\underline{J}} \left\{ \log \mathbb{P}(q|\underline{s},\underline{J}) \right\} . \tag{12.40}$$

Here (as in the ϵ-coupling method) the reference configuration is drawn from the Boltzmann distribution. In other words

$$\mathbb{E}_{\underline{s},\underline{J}}(\cdots) = \mathbb{E}_{\underline{J}} \Big\{ \frac{1}{Z_{\underline{J}}} \sum_{\underline{s}} e^{-\beta E_{\underline{J}}(\underline{s})} (\cdots) \Big\} . \tag{12.41}$$

If, as expected, $\log \mathbb{P}(q|\underline{s},\underline{J})$ concentrates in probability, one has $\mathbb{P}(q|\underline{s},\underline{J}) \doteq e^{-NV(q)}$ with high probability.

Exercise 12.11 Consider the following refined version of the toy model (12.38): $\mathbb{P}(q|\underline{s},\underline{J}) = (1 - e^{-N\Sigma(\underline{s},\underline{J})}) G_{q_0(\underline{s},\underline{J});(b_0/N\beta)}(q) + e^{-N\Sigma(\underline{s},\underline{J})} G_{q_1(\underline{s},\underline{J});(b_1/N\beta)}(q)$, where $G_{a,b}$ is a Gaussian distribution of mean a and variance b. We suppose that b_0, b_1 are constants, but $\Sigma(\underline{s},\underline{J}), q_0(\underline{s},\underline{J}), q_1(\underline{s},\underline{J})$ fluctuate as follows: when $\underline{J}$ and $\underline{s}$ are distributed according to the correct joint distribution (12.41), then $\Sigma(\underline{s},\underline{J}), q_0(\underline{s},\underline{J}), q_1(\underline{s},\underline{J})$ are independent Gaussian random variable of means respectively $\overline{\Sigma}, \overline{q}_0, \overline{q}_1$ and variances $\delta\Sigma^2/N, \delta q_0^2/N, \delta q_1^2/N$.

Assuming for simplicity that $\delta\Sigma^2 < 2\overline{\Sigma}$, compute $P(q)$ and $d(\epsilon)$ for this model. Show that the glass potential $V(q)$ is given by two arcs of parabolas:

$$V(q) = \min \left\{ \frac{(q-\overline{q}_0)^2}{2b_0} \ , \ \frac{(q-\overline{q}_1)^2}{2b_1} + \frac{1}{\beta}\,\overline{\Sigma} \right\} \tag{12.42}$$

The glass potential $V(q)$ has been computed using the replica method, only in a small number of cases, mainly fully connected p-spin glasses. Here we shall just mention the qualitative behavior that is expected on the basis of these computations. The result is summarized in Fig. 12.4. At small enough β the glass potential is convex. Increasing β one first encounters a value β_* where $V(q)$ stops to be convex. When $\beta > \beta_{\rm d} = 1/T_{\rm d}$, $V(q)$ develops a secondary minimum, at $q = q_1(\beta) > q_0(\beta)$. This secondary minimum is in fact an indication of the existence of an exponential number of clusters, such that two configurations in the same cluster typically have overlap q_1, while two configurations in distinct clusters have overlap q_0. A little thought shows that the difference between the value of the glass potential at the two minima gives the complexity: $V(q_1) - V(q_0) = T\Sigma$.

In models in which the glass potential has been computed exactly, the temperature $T_{\rm d}$ computed in this way coincides with a dramatic slowing down of the relaxational dynamics. More precisely, a properly defined relaxation time for Glauber-type dynamics is finite for $T > T_{\rm d}$ and diverges exponentially in the system size for $T < T_{\rm d}$.

12.3.6 Cloning and the complexity function

When the various clusters don't have all the same weight, the system is most appropriately described through a **complexity function**. Consider a cluster of configurations, called α. Its free-energy F_α can be defined by restricting the partition function to configurations in cluster α. One way of imposing this restriction is to chose a reference

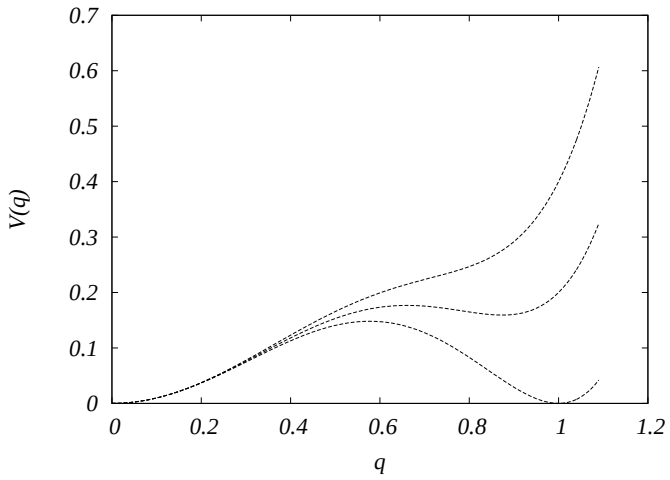

Fig. 12.4 Qualitative shapes of the glass potential $V(q)$ at various temperatures. When the temperature is very high (not shown) $V(q)$ is convex. Below $T = T_{\rm d}$, it develops a secondary minimum. The height difference between the two minima is $V(q_1) - V(q_0) = T\Sigma$. In the case shown here $q_0 = 0$ is independent of the temperature.

configuration $\underline{\sigma}_0 \in \alpha$, and restricting the Boltzmann sum to those configurations $\underline{\sigma}$ whose distance from $\underline{\sigma}_0$ is smaller than $N\delta$. In order to correctly identify clusters, one has to take $(1 - q_1)/2 < \delta < (1 - q_*)/2$, where $q_* > q_1$ is such that $V(q_*) > V(q_1)$.

Let $\mathcal{N}_\beta(f)$ be the number of clusters such that $F_\alpha = Nf$ (more precisely, this is an un-normalized measure attributing unit weight to the points F_α/N). We expect it to satisfy a large deviations principle of the form

$$\mathcal{N}_\beta(f) \doteq \exp\{N\Sigma(\beta, f)\}\,. \tag{12.43}$$

The rate function $\Sigma(\beta, f)$ is the complexity function. If clusters are defined as above, with the cut-off δ in the appropriate interval, they are expected to be disjoint up to a subset of configurations of exponentially small Boltzmann weight. Therefore the total partition function is given by:

$$Z = \sum_\alpha e^{-\beta F_\alpha} \doteq \int e^{N[\Sigma(\beta,f)-\beta f]}\,{\rm d}f \doteq e^{N[\Sigma(\beta,f_*)-\beta f_*]}\,, \tag{12.44}$$

where we applied the saddle point method as in standard statistical mechanics calculations, cf. Sec. 2.4. Here $f_* = f_*(\beta)$ solves the saddle point equation $\partial\Sigma/\partial f = \beta$.

For several reasons, it is interesting to determine the full complexity function $\Sigma(\beta, f)$, as a function of f for a given inverse temperature β. The **cloning method** is a particularly efficient (although non-rigorous) way to do this computation. Here we sketch the basic idea: several applications will be discussed in the next chapters. One begins by introducing m identical 'clones' of the initial system. These are non-interacting except for the fact that they are constrained to be in the same cluster. In

practice one can constrain all their pairwise Hamming distances to be smaller than $N\delta$, where $(1-q_1)/2 < \delta < (1-q_*)/2$. The partition function for the m clones systems is therefore

$$Z_m = {\sum_{\underline{\sigma}^{(1)},\ldots,\underline{\sigma}^{(m)}}}' \exp\Big\{ -\beta E(\underline{\sigma}^{(1)}) \cdots - \beta E(\underline{\sigma}^{(m)}) \Big\} . \tag{12.45}$$

where the prime reminds us that $\underline{\sigma}^{(1)}, \ldots \underline{\sigma}^{(m)}$ stay in the same cluster. By splitting the sum over the various clusters we have

$$Z_m = \sum_\alpha \sum_{\underline{\sigma}^{(1)} \ldots \underline{\sigma}^{(m)} \in \alpha} e^{-\beta E(\underline{\sigma}^{(1)}) \cdots - \beta E(\underline{\sigma}^{(m)})} = \sum_\alpha \Big(\sum_{\underline{\sigma} \in \alpha} e^{-\beta E(\underline{\sigma})} \Big)^m . \tag{12.46}$$

At this point we can proceed as for the calculation of the usual partition function and obtain

$$Z_m = \sum_\alpha e^{-\beta m F_\alpha} \doteq \int e^{N[\Sigma(\beta,f) - \beta m f]} \, \mathrm{d}f \doteq e^{N[\Sigma(\beta,\hat{f}) - \beta m \hat{f}]} , \tag{12.47}$$

where $f_* = f_*(\beta, m)$ solves the saddle point equation $\partial\Sigma/\partial f = \beta m$.

The free-energy density per clone of the cloned system is defined as

$$\Phi(\beta, m) = - \lim_{N\to\infty} \frac{1}{\beta m N} \log Z_m . \tag{12.48}$$

The saddle point estimate (12.47) implies that $\Phi(\beta, m)$ is related to $\Sigma(\beta, f)$ through Legendre transform:

$$\Phi(\beta, m) = f - \frac{1}{\beta m} \Sigma(\beta, f) , \qquad \frac{\partial \Sigma}{\partial f} = \beta m . \tag{12.49}$$

If we forget that m is an integer, and admit that $\Phi(\beta, m)$ can be 'continued' to non-integer m, the complexity $\Sigma(\beta, f)$ can be computed from $\Phi(\beta, m)$ by inverting this Legendre transform. The similarity to the procedure used in the replica method is not fortuitous. Notice however that replicas are introduced to deal with quenched disorder, while cloning is more general: it also applies to systems without disorder.

Exercise 12.12 In the REM, the natural definition of overlap between two configurations $i, j \in \{1, \dots, 2^N\}$ is $q_{i,j} = \mathbb{I}(i = j)$. Taking a configuration j_0 as reference, the ϵ-perturbed energy of a configuration j is $E'(\epsilon, j) = E_j - N\epsilon\mathbb{I}(j = j_0)$. (Note the extra N multiplying ϵ, introduced in order to ensure that the new ϵ-coupling term is typically extensive).

(*a*) Consider the high temperature phase where $\beta < \beta_c = 2\sqrt{\log 2}$. Show that the ϵ-perturbed system has a phase transition at $\epsilon = \frac{\log 2}{\beta} - \frac{\beta}{4}$.

(*b*) In the low temperature phase $\beta > \beta_c$, show that the phase transition takes place at $\epsilon = 0$.

Therefore in the REM the clusters exist at any β, and every cluster is reduced to one single configuration: one must have $\Sigma(\beta, f) = \log 2 - f^2$ independently of β. Show that this is compatible with the cloning approach, through a computation of the potential $\Phi(\beta, m)$:

$$\Phi(\beta, m) = \begin{cases} -\frac{\log 2}{\beta m} - \frac{\beta m}{4} & \text{for } m < \frac{\beta_c}{\beta} \\ -\sqrt{\log 2} & \text{for } m > \frac{\beta_c}{\beta} \end{cases} \tag{12.50}$$

12.4 An example: the phase diagram of the SK model

Several mean field models have been solved using the replica method. Sometimes a model may present two or more glass phases with different properties. Determining the phase diagram can be particularly challenging in these cases.

A classical example is provided by the SK model with ferromagnetically biased couplings. As in the other examples of this chapter, this is a model for N Ising spins $\underline{\sigma} = (\sigma_1, \dots, \sigma_N)$. The energy function is

$$E(\underline{\sigma}) = -\sum_{(i,j)} J_{ij}\sigma_i\sigma_j \ , \tag{12.51}$$

where (i, j) are un-ordered couples, and the couplings J_{ij} are i.i.d. Gaussian random variables with mean J_0/N and variance $1/N$. The model somehow interpolates between the Curie-Weiss model treated in Sec. 2.5.2, corresponding to $J_0 \to \infty$, and the unbiased Sherrington-Kirkpatrick model, considered in Ch. **??**, for $J_0 = 0$.

The phase diagram is plotted in terms of two parameters: the ferromagnetic bias J_0, and the temperature T. Depending on their values, the system is found in one of four phases, cf. Fig. 12.5: paramagnetic (P), ferromagnetic (F), symmetric spin glass (SG) and mixed ferromagnetic spin glass (F-SG). A simple characterization of these four phases is obtained in terms of two quantities: the average magnetization and overlap. In order to define them, we must first observe that, since $E(\underline{\sigma}) = E(-\underline{\sigma})$, in the present model $\langle\sigma_i\rangle = 0$ identically for all values of J_0, and T. In order to break this symmetry, we may add a magnetic field term $-B\sum_i \sigma_i$ and let $B \to 0$ after the thermodynamic limit. We then define

$$m = \lim_{B\to 0+} \lim_{N\to\infty} \mathbb{E}\langle\sigma_i\rangle_B \, , \qquad \overline{q} = \lim_{B\to 0+} \lim_{N\to\infty} \mathbb{E}(\langle\sigma_i\rangle_B^2) \, , \tag{12.52}$$

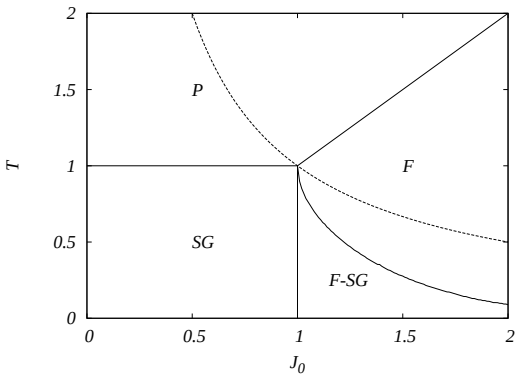

Fig. 12.5 Phase diagram of the SK model in zero magnetic field. When the temperature T and the ferromagnetic bias J_0 are varied, four distinct phases are encountered: paramagnetic (P), ferromagnetic (F), spin glass (SG) and mixed ferromagnetic-spin glass (F-SG). The full lines separate these various phases. The dashed line is the location of the Nishimori temperature.

(which don't depend on i because the coupling distribution is invariant under a permutation of the sites). In the P phase one has $m = 0, \overline{q} = 0$; in the SG phase $m = 0, \overline{q} > 0$, and in the F and F-SG phases one has $m > 0, \overline{q} > 0$.

A more complete description is obtained in terms of the overlap distribution $P(q)$. Because of the symmetry under spin inversion mentioned above, $P(q) = P(-q)$ identically. The qualitative shape of $P(q)$ in the thermodynamic limit is shown in Fig. 12.6. In the P phase it consists of a single delta function with unit weight at $q = 0$: two independent configurations drawn from the Boltzmann distribution have, with high probability, overlap close to 0. In the F phase, it is concentrated on two symmetric values $q(J_0, T) > 0$ and $-q(J_0, T) < 0$, each carrying weight one half. We can summarize this behavior by saying that a random configuration drawn from the Boltzmann distribution is found, with equal probability, in one of two different states. In the first one the local magnetizations are $\{m_i\}$, in the second one they are $\{-m_i\}$. If one draws two independent configurations, they fall in the same state (corresponding to the overlap value $q(J_0, T) = N^{-1} \sum_i m_i^2$) or in opposite states (overlap $-q(J_0, T)$) with probability 1/2. In the SG phase the support of $P(q)$ is a symmetric interval $[-q_{\max}, q_{\max}]$, with $q_{\max} = q_{\max}(J_0, T)$. Finally, in the F-SG phase the support is the union of two intervals $[-q_{\max}, -q_{\min}]$ and $[q_{\min}, q_{\max}]$. Both in the SG and F-SG phases, the presence of a whole range of overlap values carrying non-vanishing probability, suggests the existence of a multitude of quasi-states (in the sense discussed in the previous Section).

In order to remove the degeneracy due to the symmetry under spin inversion, one sometimes define an asymmetric overlap distribution by adding a magnetic field terms, and taking the thermodynamic limit as in Eq. (12.52):

$$P_+(q) = \lim_{B \to 0+} \lim_{N \to \infty} P_B(q)\,. \tag{12.53}$$

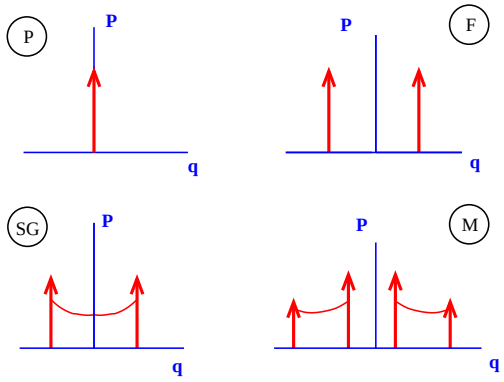

Fig. 12.6 The typical shape of the $P(q)$ function in each of the four phases of the SK model ferromagnetically biased couplings.

Somewhat surprisingly, it turns out that $P_+(q) = 0$ for $q < 0$, while $P_+(q) = 2P(q)$ for $q > 0$. In other words $P_+(q)$ is equal to the distribution of the *absolute value* of the overlap.

Exercise 12.13 Consider the Curie-Weiss model in a magnetic field, cf. Sec. 2.5.2. Draw the phase diagram and compute the asymptotic overlap distribution. Discuss its qualitative features for different values of the temperature and magnetic field.

A few words for the reader interested in how one derives this diagram: Some of the phase boundaries were already derived using the replica method in Exercise **??**. The boundary P-F is obtained by solving the RS equation (**??**) for q, ω, $\mathtt{x}$. The P-SG and F-M lines are obtained by the AT stability condition (**??**). Deriving the phase boundary between the SG and F-SG phases is much more challenging, because it separates glassy phases, therefore it cannot be derived within the RS solution. It is known to be approximately vertical, but there is no simple expression for it. The Nishimori temperature is deduced from the condition (12.7): $T_{\rm N} = 1/J_0$, and the line $T = 1/J_0$ is usually called 'Nishimori line'. The internal energy per spin on this line is $U/N = -J_0/2$. Notice that the line does not enter any of the glass phases, as we know from general arguments.

An important aspect of the SK model is that the appearance of the glass phase on the lines separating P from SG on the one hand, and F from F-SG on the other hand is a continuous transition. Therefore it is associated with the divergence of the non-linear susceptibility $\chi_{\rm SG}$. The following exercise, reserved to the replica aficionados, sketches the main lines of the argument showing this.

Exercise 12.14 Let us see how to compute the non-linear susceptibility of the SK model, $\chi_{\rm SG} = \frac{\beta^2}{N} \sum_{i \neq j} \left(\langle \sigma_i \sigma_j \rangle - \langle \sigma_i \rangle \langle \sigma_j \rangle \right)^2$, with the replica method Show that:

$$\begin{aligned} \chi_{\rm SG} &= \lim_{n \to 0} \frac{\beta^2}{N} \sum_{i \neq j} \left(\binom{n}{2}^{-1} \sum_{(ab)} \langle \sigma_i^a \sigma_i^b \sigma_j^a \sigma_j^b \rangle - \binom{n}{3}^{-1} \sum_{(abc)} \langle \sigma_i^a \sigma_i^b \sigma_j^a \sigma_j^c \rangle \right. \\ &\qquad \left. + \binom{n}{4}^{-1} \sum_{(abcd)} \langle \sigma_i^a \sigma_i^b \sigma_j^c \sigma_j^d \rangle \right) \\ &= N \lim_{n \to 0} \int e^{-NG(Q,\lambda)} A(Q) \prod_{(ab)} (\mathrm{d}Q_{ab} \mathrm{d}\lambda_{ab}) \; , \end{aligned} \tag{12.54}$$

where we follow the notations of (**??**), the sum over $(a_1 a_2 \dots a_k)$ is understood to run over all the k-uples of distinct replica indices, and

$$A(Q) \equiv \binom{n}{2}^{-1} \sum_{(ab)} Q_{ab}^2 - \binom{n}{3}^{-1} \sum_{(abc)} Q_{ab} Q_{ac} + \binom{n}{4}^{-1} \sum_{(abcd)} Q_{ab} Q_{cd} \, . \tag{12.55}$$

Analyze the divergence of $\chi_{\rm SG}$ along the following lines: The leading contribution to (12.54) should come from the saddle point and be given, in the high temperature phase, by $A(Q_{ab} = q)$ where $Q_{ab} = q$ is the RS saddle point. However this contribution clearly vanishes when one takes the $n \to 0$ limit. One must thus consider the fluctuations around the saddle point. Each of the terms like $Q_{ab} Q_{cd}$ in $A(Q)$ gives a factor $\frac{1}{N}$ time the appropriate matrix element of the inverse of the Hessian matrix. When this Hessian matrix is non-singular, these elements are all finite and one obtains a finite result in the $N \to \infty$ limit. (The $1/N$ cancels the factor N in (12.54)). When one reaches the AT instability line, the elements of the inverse of the Hessian matrix diverge, and therefore $\chi_{\rm SG}$ also diverges.

Notes

Lattice gas models of atomic systems, such as those discussed in the first two exercises, are discussed in statistical physics textbooks, see for instance (Ma, 1985). The simple model of glasses of exercise 12.3 was introduced and solved with the cavity method by (Biroli and Mézard, 2002).

The order parameter for spin glasses defined by (Edwards and Anderson, 1975) is a dynamic order parameter which captures the long time persistence of the spins. The static definition that we have introduced here should give the same result as the original dynamical one (although of course we have no proof of this statement in general). A review on the simulations of the Edwards-Anderson model can be found in (Marinari, Parisi and Ruiz-Lorenzo, 1997).

Mathematical results on mean field spin glasses are found in the book (Talagrand, 2003). A short recent survey is provided by (Guerra, 2005).

Diluted spin glasses were introduced in (Viana and Bray, 1985).

The implications of the gauge transformation were derived by Hidetoshi Nishimori and his coworkers, and are explained in details in his book (Nishimori, 2001).

The notion of pure states in phase transitions, and the decomposition of Gibbs measures into pure states, is discussed in the book (Georgii, 1988). We shall discuss this topic further in Chapter **??**.

The divergence of the spin glass susceptibility is specially relevant because this susceptibility can be measured in zero field. The experiments of (Monod and Bouchiat, 1982) present evidence of a divergence. This supports the existence of a spin glass transition in real (three dimensional) spin glasses in zero magnetic field, at non-zero temperature.

The existence of two transition temperatures $T_c < T_d$ was first discussed by Kirkpatrick, Thirumalai and Wolynes (Kirkpatrick and Wolynes, 1987; Kirkpatrick and Thirumalai, 1987), who pointed out the relevance to the theory of structural glasses. In particular, (Kirkpatrick and Thirumalai, 1987) discusses the case of the p-spin glass. A review of this line of approach to structural glasses, and particularly its relevance to dynamical effects, is (Bouchaud, Cugliandolo, Kurchan and Mézard, 1997).

The ϵ-coupling method was introduced in (Caracciolo, Parisi, Patarnello and Sourlas, 1990). The idea of cloning in order to study the complexity function is due to Monasson (Monasson, 1995). Its application to studies of the glass transition without quenched disorder was developed in (Mézard and Parisi, 1999).

The glass potential method was introduced in (Franz and Parisi, 1995).

14
Belief propagation

Consider the ubiquitous problem of computing marginals of a graphical model with N variables $\underline{x} = (x_1, \dots, x_N)$ taking values in a finite alphabet $\mathcal{X}$. The naive algorithm, summing over all configurations, takes a time of order $|\mathcal{X}|^N$. The complexity can be reduced dramatically when the underlying factor graph has some special structure. One extreme case is that of tree factor graphs. On trees, marginals can be computed in a number of operations which grows linearly with N. This can be done through a 'dynamic programming' procedure that recursively sums over all variables starting from the leaves and progressing towards the 'center' of the tree.

Remarkably, such a recursive procedure can be recast as a distributed 'message passing' algorithm. Message passing algorithms operate on 'messages' associated with edges of the factor graph, and update them recursively through local computations done at the vertices of the graph. The update rules that yield exact marginals on trees have been discovered independently in several different contexts: statistical physics (under the name 'Bethe Peierls approximation'), coding theory ('sum-product' algorithm), and artificial intelligence ('belief propagation' - BP). Here we will adopt the artificial intelligence terminology.

This chapter gives a detailed presentation of BP, and more generally message passing procedures, which provide one of the main building blocks that we use throughout the rest of the book. It is therefore important that the reader has a good understanding of it.

It is straightforward to prove that BP exactly computes marginals on tree factor graphs. However, it was found only recently that it can be extremely effective on loopy graphs as well. One of the basic intuitions behind this success is that BP, being a local algorithm, should be successful whenever the underlying graph is 'locally' a tree. Such factor graphs appear frequently, for instance in error correcting codes, and BP turns out to be very powerful in this context. However, even in such cases, its application is limited to distributions such that far apart variables become approximately uncorrelated. The onset of long range correlations, typical of the occurrence of a phase transition, generically leads to poor performances of BP. We shall see several applications of this idea in the next chapters.

We introduce the basic ideas in Section 14.1 by working out a couple of simple examples. The general BP equations are stated in Section 14.2, which also shows how they provide exact results on tree factor graphs. Section 14.3 describes an alternative message passing procedure, the max-product (equivalently, min-sum) algorithm, which can be used in optimization problems. In Section 14.4 we discuss the use of BP in graphs with loops. In the study of random constraint satisfaction problems, BP

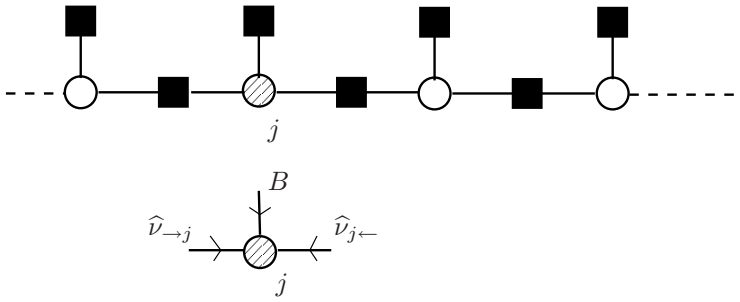

Fig. 14.1 Top: the factor graph of the one-dimensional Ising model in an external field. Bottom: the three messages arriving on site j describe the contributions to the probability distribution of σ_j, due to the left chain ($\widehat{\nu}_{\to j}$), to the right chain ($\widehat{\nu}_{j\leftarrow}$) and to the external field B.

messages become random variables. The study of their distribution provides a large amount of information on such instances and can be used to characterize the corresponding phase diagram. The time evolution of these distributions is known under the name of density evolution, while their fixed point analysis is the replica symmetric cavity method. Both are explained in Section 14.6.

14.1 Two examples

14.1.1 Example 1: Ising chain

Consider the ferromagnetic Ising model on a line. The variables are Ising spins $(\sigma_1, \dots, \sigma_N) = \underline{\sigma}$, with $\sigma_i \in \{+1, -1\}$ and their joint distribution takes Boltzmann's form

$$\mu_\beta(\underline{\sigma}) = \frac{1}{Z}\, e^{-\beta E(\underline{\sigma})}\,, \qquad E(\underline{\sigma}) = -\sum_{i=1}^{N-1} \sigma_i\sigma_{i+1} - B\sum_{i=1}^{N} \sigma_i\,. \tag{14.1}$$

The corresponding factor graph is shown in Figure 14.1.1.

Let us now compute the marginal probability distribution $\mu(\sigma_j)$ of spin σ_j. We shall introduce three 'messages' arriving on spin j as the contributions to $\mu(\sigma_j)$ coming from each of the function nodes which are connected to i. More precisely, let us define

$$\widehat{\nu}_{\to j}(\sigma_j) = \frac{1}{Z_{\to j}} \sum_{\sigma_1 \dots \sigma_{j-1}} \exp\left\{\beta \sum_{i=1}^{j-1} \sigma_i\sigma_{i+1} + \beta B \sum_{i=1}^{j-1} \sigma_i\right\}\,,$$
$$\widehat{\nu}_{j\leftarrow}(\sigma_j) = \frac{1}{Z_{j\leftarrow}} \sum_{\sigma_{j+1} \dots \sigma_N} \exp\left\{\beta \sum_{i=j}^{N-1} \sigma_i\sigma_{i+1} + \beta B \sum_{i=j+1}^{N} \sigma_i\right\}\,. \tag{14.2}$$

Messages are understood to be probability distributions and thus normalized. In the present case, the constants $Z_{\to j}$, $Z_{j\leftarrow}$ are set by the conditions $\widehat{\nu}_{\to j}(+1)+\widehat{\nu}_{\to j}(-1) = 1$, and $\widehat{\nu}_{j\leftarrow}(+1)+\widehat{\nu}_{j\leftarrow}(-1) = 1$. In the following, when dealing with normalized distributions, we shall avoid writing explicitly the normalization constants and use the symbol

$\cong$ to denote 'equality up to a normalization'. With this notation, the first of the above equations can be rewritten as

$$\widehat{\nu}_{\rightarrow j}(\sigma_j) \cong \sum_{\sigma_1 \dots \sigma_{j-1}} \exp\left\{ \beta \sum_{i=1}^{j-1} \sigma_i \sigma_{i+1} + \beta B \sum_{i=1}^{j-1} \sigma_i \right\} . \tag{14.3}$$

By rearranging the summation over spins σ_i, $i \neq j$, the marginal $\mu(\sigma_j)$ can be written as:

$$\mu(\sigma_j) \cong \widehat{\nu}_{\rightarrow j}(\sigma_j)\, e^{\beta B \sigma_j}\, \widehat{\nu}_{j\leftarrow}(\sigma_j) . \tag{14.4}$$

In this expression we can interpret each of the three factors as a 'message' sent to j from each of the three function nodes connected to the variable j. Each message coincides with the marginal distribution of σ_j in a modified graphical model. For instance, $\widehat{\nu}_{\rightarrow j}(\sigma_j)$ is the distribution of σ_j in the graphical model obtained by removing all the factor nodes adjacent to j, except the one on its left (cf. Fig. 14.1.1).

This decomposition is interesting because the various messages can be computed iteratively. Consider for instance $\widehat{\nu}_{\rightarrow i+1}$. It is expressed in terms of $\widehat{\nu}_{\rightarrow i}$ as:

$$\widehat{\nu}_{\rightarrow i+1}(\sigma) \cong \sum_{\sigma'} \widehat{\nu}_{\rightarrow i}(\sigma')\; e^{\beta\sigma'\sigma + \beta B \sigma'} . \tag{14.5}$$

Furthermore, $\widehat{\nu}_{\rightarrow 1}$ is the uniform distribution over $\{+1, -1\}$: $\widehat{\nu}_{\rightarrow 1}(\sigma) = \frac{1}{2}$ for $\sigma = \pm 1$. Equation (14.5) allows to compute all the messages $\widehat{\nu}_{\rightarrow i}$, $i \in \{1, \dots, N\}$, in $O(N)$ operations. A similar procedure yields $\widehat{\nu}_{i\leftarrow}$ starting from the uniform distribution $\widehat{\nu}_{N\leftarrow}$ and computing recursively $\widehat{\nu}_{i-1\leftarrow}$ from $\widehat{\nu}_{i\leftarrow}$. Finally, Eq. (14.4) can be used to compute all the marginals $\mu(\sigma_j)$ in linear time.

All the messages are distributions over binary variables and can thus be parameterized by a single real number. One popular choice for such a parameterization is to use the log-likelihood ratio[1]

$$u_{\rightarrow i} \equiv \frac{1}{2\beta} \log \frac{\widehat{\nu}_{\rightarrow i}(+1)}{\widehat{\nu}_{\rightarrow i}(-1)} . \tag{14.6}$$

In statistical physics terms $u_{\rightarrow i}$ is an 'effective (or local) magnetic field': $\widehat{\nu}_{\rightarrow i}(\sigma) \cong e^{\beta u_{\rightarrow i}\sigma}$. Using this definition (and noticing that it implies $\widehat{\nu}_{\rightarrow i}(\sigma) = \frac{1}{2}(1+\sigma \tanh(\beta u_{\rightarrow i}))$), Eq. (14.5) becomes:

$$u_{\rightarrow i+1} = f\left(u_{\rightarrow i} + B\right) , \tag{14.7}$$

where the function $f(x)$ is defined as

$$f(x) = \frac{1}{\beta} \mathrm{atanh}\left[\tanh(\beta)\tanh(\beta x)\right] . \tag{14.8}$$

The mapping $u \mapsto f(u+B)$ is differentiable with derivative bounded by $\tanh\beta < 1$. Therefore the fixed point equation $u = f(u+B)$ has a unique solution u_*, and $u_{\rightarrow i}$

[1]Notice that our definition differs by a factor $1/2\beta$ from the standard log-likelihood definitions in statistics. This factor is introduced to make contact with statistical physics definitions.

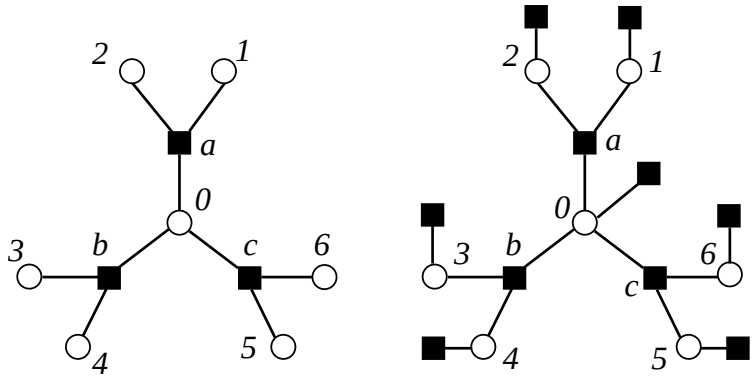

Fig. 14.2 Left: A simple parity check code with 7 variables and 3 checks. Right: the factor graph corresponding to the problem of finding the sent codeword, given a received message.

goes to u_* when $i \to \infty$. Consider a very long chain, and a node in the bulk $j \in [\varepsilon N, (1-\varepsilon)N]$. Then, as $N \to \infty$, both $u_{\to j}$ and $u_{j\leftarrow}$ converge to u^*, so that $\langle \sigma_j \rangle \to \tanh[\beta(2u^* + B)]$. This is the bulk magnetization. If on the other hand we consider a spin on the boundary we get a smaller magnetization $\langle \sigma_1 \rangle = \langle \sigma_N \rangle \to \tanh[\beta(u^* + B)]$.

Exercise 14.1 Use the recursion (14.7) to show that, when N and j go to infinity, $\langle \sigma_j \rangle = M + O(\lambda^j, \lambda^{N-j})$ where $M = \tanh(2u_* + B)$ and $\lambda = f'(u_* + B)$. Compare this with the treatment of the one-dimensional Ising model in Sec. 2.5.

The above method can be generalized to the computation of joint distributions of two or more variables. Consider for instance the joint distribution $\mu(\sigma_j, \sigma_k)$, for $k > j$. Since we already know how to compute the marginal $\mu(\sigma_j)$, it is sufficient to consider the conditional distribution $\mu(\sigma_k|\sigma_j)$. For each of the two values of σ_j, the conditional distribution of $\sigma_{j+1}, \cdots, \sigma_N$ takes a form analogous to Eq. (14.1) but with σ_j fixed. Therefore, the marginal $\mu(\sigma_k|\sigma_j)$ can be computed through the same algorithm as before. The only difference is in the initial condition that becomes $\widehat{\nu}_{\to j}(+1) = 1$, $\widehat{\nu}_{\to j}(-1) = 0$ (if we condition on $\sigma_j = +1$) and $\widehat{\nu}_{\to j}(+1) = 0$, $\widehat{\nu}_{\to j}(-1) = 1$ (if we condition on $\sigma_j = -1$).

Exercise 14.2 Compute the correlation function $\langle \sigma_j \sigma_k \rangle$, when $j, k \in [N\varepsilon, N(1-\varepsilon)]$ and $N \to \infty$. Check that, when $B = 0$, $\langle \sigma_j \sigma_k \rangle = (\tanh \beta)^{|j-k|}$. Find a simpler derivation of this last result.

14.1.2 Example 2: a tree-parity-check code

Our second example deals with a decoding problem. Consider the simple linear code whose factor graph is reproduced in Fig. 14.1.2, left frame. It has block-length $N = 7$

and codewords satisfy the 3 parity check equations:

$$x_0 \oplus x_1 \oplus x_2 = 0\,, \tag{14.9}$$
$$x_0 \oplus x_3 \oplus x_4 = 0\,, \tag{14.10}$$
$$x_0 \oplus x_5 \oplus x_6 = 0\,. \tag{14.11}$$

One of the codewords is sent through a BSC(p). Assume that the received message is $\underline{y} = (1,0,0,0,0,1,0)$. The conditional distribution for $\underline{x}$ to be the transmitted codeword, given the received $\underline{y}$, takes the usual form $\mu_{\underline{y}}(\underline{x}) = \mathbb{P}(\underline{x}|\underline{y})$:

$$\mu_{\underline{y}}(\underline{x}) \cong \mathbb{I}(x_0 \oplus x_1 \oplus x_2 = 0)\mathbb{I}(x_0 \oplus x_3 \oplus x_4 = 0)\mathbb{I}(x_0 \oplus x_5 \oplus x_6 = 0) \prod_{i=0}^{6} Q(y_i|x_i)\ ,$$

where $Q(0|0) = Q(1|1) = 1-p$ and $Q(1|0) = Q(0|1) = p$. The corresponding factor graph is drawn in Fig. 14.1.2, right frame.

In order to implement symbol MAP decoding, cf. Ch. 6, we need to compute the marginal distribution of each bit. The computation is straightforward but it is illuminating to recast it as a message passing procedure, similar to the one in the Ising chain example. Consider for instance bit x_0. We start from the boundary. In the absence of the check a, the marginal of x_1 would be $\nu_{1\to a} = (1-p, p)$ (we use here the convention of writing distributions $\nu(x)$ over a binary variable as two dimensional vectors $(\nu(0), \nu(1))$). This is interpreted as a message sent from variable 1 to check a.

Variable 2 sends an analogous message $\nu_{2\to a}$ to a (in the present example, this happens to be equal to $\nu_{1\to a}$). Knowing these two messages, we can compute the contribution to the marginal probability distribution of variable x_0 coming from the part of the factor graph containing the whole branch connected to x_0 through the check a:

$$\widehat{\nu}_{a\to 0}(x_0) \cong \sum_{x_1,x_2} \mathbb{I}(x_0 \oplus x_1 \oplus x_2 = 0)\ \nu_{1\to a}(x_1)\nu_{2\to a}(x_2)\ . \tag{14.12}$$

Clearly, $\widehat{\nu}_{a\to 0}(x_0)$ is the marginal distribution of x_0 in the modified factor graph that does not include either factor node b or c, and in which the received symbol y_0 has been erased. This is analogous to the messages $\widehat{\nu}_{\to j}(\sigma_j)$ used in the Ising chain example. The main difference is that the underlying factor graph is no longer a line, but a tree. As a consequence, the recursion (14.12) is no longer linear in the incoming messages. Using the rule (14.12), and analogous ones for $\widehat{\nu}_{b\to 0}(x_0)$, $\widehat{\nu}_{c\to 0}(x_0)$, we obtain:

$$\begin{aligned}
\widehat{\nu}_{a\to 0} &= (p^2 + (1-p)^2,\ 2p(1-p))\,,\\
\widehat{\nu}_{b\to 0} &= (p^2 + (1-p)^2,\ 2p(1-p))\,,\\
\widehat{\nu}_{c\to 0} &= (2p(1-p),\ p^2 + (1-p)^2)\,.
\end{aligned}$$

The marginal probability distribution of the variable x_0 is finally obtained by taking into account the contributions of each subtree, together with the channel output for bit x_0:

$$\begin{aligned}
\mu(x_0) &\cong Q(y_0|x_0)\,\widehat{\nu}_{a\to 0}(x_0)\widehat{\nu}_{b\to 0}(x_0)\widehat{\nu}_{c\to 0}(x_0)\\
&\cong \big(2p^2(1-p)[p^2 + (1-p)^2]^2,\ \ 4p^2(1-p)^3[p^2 + (1-p)^2]\big)
\end{aligned}$$

In particular, the MAP symbol decoding of the symbol x_0 is always $x_0 = 0$ in this case, for any $p < 1/2$.

An important fact emerges from this simple calculation. Instead of performing a summation over $2^7 = 128$ configurations, we were able to compute the marginal at x_0 doing 6 summations (one for every factor node a, b, c and for every value of x_0), each one over 2 summands, cf. Eq. (14.12). Such complexity reduction was achieved by merely rearranging the order of sums and multiplications in the marginal computation.

Exercise 14.3 Show that the message $\nu_{0\to a}(x_0)$ is equal to $(1/2, 1/2)$, and deduce that $\mu(x_1) \cong ((1-p), p)$.

14.2 Belief Propagation on tree graphs

We shall define belief propagation and analyze it in the simplest possible setting: tree graphical models. In this case it solves several computational problems in an efficient and distributed fashion.

14.2.1 Three problems

Let us consider a graphical model such that the associated factor graph is a tree (we shall call it a **tree-graphical model**). We use the same notations as in Sec. 9.1.1. The model describes N random variables $(x_1, \dots, x_N) \equiv \underline{x}$ taking values in a finite alphabet $\mathcal{X}$, whose joint probability distribution has the form

$$\mu(\underline{x}) = \frac{1}{Z} \prod_{a=1}^{M} \psi_a(\underline{x}_{\partial a})\,. \tag{14.13}$$

where $\underline{x}_{\partial a} \equiv \{x_i \,|\, i \in \partial a\}$. The set $\partial a \subseteq [N]$, of size $|\partial a|$, contains all variables involved in constraint a. We always use indices $i, j, k, \dots$ for the variables and $a, b, c, \dots$ for the function nodes. The set of indices ∂i involves all function nodes a connected to i.

When the factor graph has no loop the following are among the basic problems that can be solved efficiently with a message-passing procedure:

1. Compute the marginal distributions of one variable, $\mu(x_i)$, or the joint distribution of a small number of variables.
2. Sample from $\mu(\underline{x})$, i.e. draw independent random configurations $\underline{x}$ with distribution $\mu(\underline{x})$.
3. Compute the partition function Z, or equivalently, in statistical physics language, the free-entropy $\log Z$.

These three tasks can be accomplished using belief propagation, which is the obvious generalization of the procedure exemplified in the previous section.

14.2.2 The BP equations

Belief propagation is an iterative 'message passing' algorithm. The basic variables on which it acts are messages associated with directed edges on the factor graph. For each

edge (i, a) (where i is a variable node and a a function node) there exist, at the t-th iteration, two messages $\nu_{i\to a}^{(t)}$ and $\widehat{\nu}_{a\to i}^{(t)}$. Messages take values in the space of probability distributions over the single variable space $\mathcal{X}$. For instance, $\nu_{i\to a}^{(t)} = \{\nu_{i\to a}^{(t)}(x_i) : x_i \in \mathcal{X}\}$, with $\nu_{i\to a}^{(t)}(x_i) \geq 0$ and $\sum_{x_i} \nu_{i\to a}^{(t)}(x_i) = 1$.

In tree-graphical models, the messages converge when $t \to \infty$ to fixed point values (see Theorem 14.1). These coincide with single variable marginals in modified graphical models, as we saw in the two examples of the previous section. More precisely $\nu_{i\to a}^{(\infty)}(x_i)$ is the marginal distribution of variable x_i in a modified graphical model which does not include the factor a (i.e. the product in Eq. (14.13) does not include a). Analogously $\widehat{\nu}_{a\to i}^{(\infty)}(x_i)$ is the distribution of x_i in a graphical model where all factors in ∂i, except a, have been erased.

Messages are updated through local computations at the nodes of the factor graph. By *local* we mean that a given node updates the outgoing messages on the basis of incoming ones at the previous iterations. This is a characteristic feature of message passing algorithms, while different algorithms in this family differ in the precise form of the update equations. The **belief propagation (BP)**, or **sum-product** update rules, are:

$$\nu_{j\to a}^{(t+1)}(x_j) \cong \prod_{b\in\partial j\setminus a} \widehat{\nu}_{b\to j}^{(t)}(x_j)\,, \tag{14.14}$$

$$\widehat{\nu}_{a\to j}^{(t)}(x_j) \cong \sum_{\underline{x}_{\partial a\setminus j}} \psi_a(\underline{x}_{\partial a}) \prod_{k\in\partial a\setminus j} \nu_{k\to a}^{(t)}(x_k)\ . \tag{14.15}$$

It is understood that, when $\partial j\setminus a$ is an empty set, $\nu_{j\to a}(x_j)$ is the uniform distribution. Similarly, if $\partial a \setminus j$ is empty, then $\widehat{\nu}_{a\to j}(x_j) = \psi_a(x_j)$. A pictorial illustration of these rules is provided in Fig. 14.2.2. A BP fixed point is a set of t-independent messages $\nu_{i\to a}^{(t)} = \nu_{i\to a}$, $\widehat{\nu}_{a\to i}^{(t)} = \widehat{\nu}_{a\to i}$ which satisfy Eqs. (14.14), (14.15). From these one obtains $2|\mathcal{E}|$ equations (one equation for each directed edge of the factor graph) relating $2|\mathcal{E}|$ messages. We will often refer to these fixed point conditions as to the **BP equations**.

After t iterations, one can estimate the marginal distribution $\mu(x_i)$ of variable i using the set of *all* incoming messages. The BP estimate is:

$$\nu_i^{(t)}(x_i) \cong \prod_{a\in\partial i} \widehat{\nu}_{a\to i}^{(t-1)}(x_i)\,. \tag{14.16}$$

In writing the update rules, we are assuming that the update is done in parallel at all the variable nodes, then in parallel at all function nodes and so on. Clearly, in this case, the iteration number must be incremented either at variable nodes or at factor nodes, but not necessarily at both. This is what happens in Eqs. (14.14), (14.15). Other update schedules are possible and sometimes useful. For the sake of simplicity we shall however stick to the parallel one.

In order to fully define the algorithm, we need to specify an initial condition. It is a widespread habit to set initial messages to the uniform distribution over $\mathcal{X}$ (i.e. $\nu_{i\to a}^{(0)}(x_i) = 1/|\mathcal{X}|$). On the other hand, it can be useful to explore several distinct

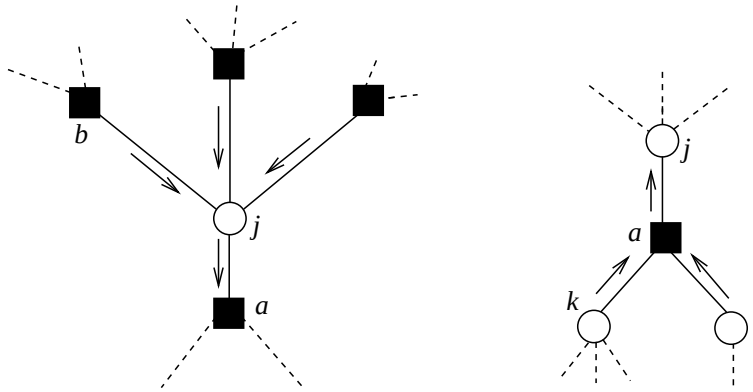

Fig. 14.3 Left: portion of the factor graph involved in the computation of $\nu_{j\to a}^{(t+1)}(x_j)$. This message is a function of the 'incoming messages' $\widehat{\nu}_{b\to j}^{(t)}(x_j)$, with $b \neq a$. Right: portion of the factor graph involved in the computation of $\widehat{\nu}_{a\to j}^{(t)}(x_j)$. This message is a function of the 'incoming messages' $\nu_{k\to a}^{(t)}(x_k)$, with $k \neq j$.

(random) initial conditions. This can be done defining some probability measure P over the space $\mathfrak{M}(\mathcal{X})$ of distributions over $\mathcal{X}$ (i.e. the $|\mathcal{X}|$-dimensional simplex) and taking $\nu_{i\to a}^{(0)}(\,\cdot\,)$ as i.i.d. random variables with distribution P.

The BP algorithm can be applied to any graphical model, irrespective of whether the factor graph is a tree or not. One possible version of the algorithm is as follows:

BP (Graphical model (G, ψ), Accuracy ϵ, Iterations $t_{\max}$)

Initialize BP messages as i.i.d. random variables with distribution P ;
For $t \in \{0, \dots, t_{\max}\}$
  For each $(j, a) \in E$
    Compute the new value of $\widehat{\nu}_{a\to j}$ using Eq. (14.15);
  For each $(j, a) \in E$
    Compute the new value of $\nu_{j\to a}$ using Eqs. (14.14);
  Let Δ be the maximum message change;
  If $\Delta < \epsilon$ return current messages;
End-For;
Return 'Not Converged';

Among all message passing algorithms, BP is uniquely characterized by the property of computing exact marginals on tree-graphical models.

Theorem 14.1. (BP is exact on trees) *Consider a tree-graphical model with diameter t_* (which means that t_* is the maximum distance between any two variable nodes). Then*

1. *Irrespective of the initial condition, the BP update (14.14), (14.15) converges after at most t_* iterations. In other words, for any edge (ia), and any $t > t_*$*

$\nu^{(t)}_{i\to a} = \nu^*_{i\to a}$, $\widehat{\nu}^{(t)}_{a\to i} = \widehat{\nu}^*_{a\to i}$.

2. *The fixed point messages provide the exact marginals: for any variable node i, and any $t > t_*$, $\nu^{(t)}_i(x_i) = \mu(x_i)$.*

Proof: As exemplified in the previous section, on tree factor graphs BP is just a clever way to organize the sum over configurations to compute marginals. In this sense the theorem is obvious.

Let us sketch a formal proof, leaving a few details to the reader. Given a directed edge $i \to a$ between a variable i and a factor node a, we define $\mathbb{T}(i \to a)$ as the sub-tree rooted on this edge. This is the subtree containing all nodes w which can be connected to i by a non-reversing path[2] which does not include the edge (i, a). Let $t_*(i \to a)$ be the *depth* of $\mathbb{T}(i \to a)$ (the maximal distance from a leaf to i).

We will show that, for any number of iterations $t > t_*(i \to a)$, the message $\nu^{(t)}_{i\to a}$ coincides with the marginal distribution of the root variable with respect to the graphical model $\mathbb{T}(i \to a)$. In other words, for tree graphs the interpretation of BP messages in terms of modified marginals is correct.

This claim is proved by induction on the tree depth $t_*(i \to a)$. The base step of the induction is trivial: $\mathbb{T}(i \to a)$ is the graph formed by the unique node i. By definition, for any $t \geq 1$, $\nu^{(t)}_{i\to a}(x_i) = 1/|\mathcal{X}|$ is the uniform distribution, which coincides with the marginal of the trivial graphical model associated to $\mathbb{T}(i \to a)$.

The induction step is easy as well. Assuming the claim to be true for $t_*(i \to a) \leq \tau$, one has to show that it holds when $t_*(i \to a) = \tau + 1$. To this end, take any $t > \tau + 1$ and compute $\nu^{(t+1)}_{i\to a}(x_i)$ using Eqs. (14.14), (14.15) in terms of messages $\nu^{(t)}_{j\to b}(x_j)$ in the subtrees for $b \in \partial i \setminus a$ and $j \in \partial b \setminus i$. By the induction hypothesis, and since the depth of the sub-tree $T(j \to b)$ is at most τ, $\nu^{(t)}_{j\to b}(x_j)$ is the root marginal in such a subtree. It turns out that, combining the marginals at roots of subtrees $\mathbb{T}(j \to b)$ using Eqs. (14.14), (14.15), one obtains the marginal at the root of $\mathbb{T}(i \to a)$. This proves the claim. □

14.2.3 Correlations and energy

The use of BP is not limited to computing one variable marginals. Suppose we want to compute the joint probability distribution $\mu(x_i, x_j)$ of two variables x_i and x_j. Since BP already enables to compute $\mu(x_i)$, this task is equivalent to computing the conditional distribution $\mu(x_j \mid x_i)$. Given a model that factorizes as in Eq. (14.13), the conditional distribution of $\underline{x} = (x_1, \dots, x_N)$ given $x_i = x$ takes the form

$$\mu(\underline{x}|x_i = x) \cong \prod_{a=1}^{M} \psi_a(\underline{x}_{\partial a})\, \mathbb{I}(x_i = x)\,. \tag{14.17}$$

In other words, it is sufficient to add to the original graph a new function node of degree 1 connected to variable node i, which fixes $x_i = x$. One can then run BP on

[2]A **non-reversing path** on a graph $\mathcal{G}$ is a sequence of vertices $\omega = (j_0, j_1, \dots, j_n)$, such that (j_s, j_{s+1}) is an edge for any $s \in \{0, \dots, n-1\}$, and $j_{s-1} \neq j_{s+1}$ for $s \in \{1, \dots, n-1\}$.

the modified factor graph and obtain estimates $\nu_j^{(t)}(x_j|x_i = x)$ for the conditional marginal of x_j.

This strategy is easily generalized to the joint distribution of any number m of variables. The complexity grows however exponentially in the number of variables involved, since we have to condition over $|\mathcal{X}|^{m-1}$ possible assignments.

Happily, for tree-graphical models, the marginal distribution of any number of variables admits an explicit expression in terms of messages. Let F_R be a subset of function nodes, V_R be the subset of variable nodes adjacent to F_R, R the induced subgraph, and $\underline{x}_R$ the corresponding variables. Without loss of generality, we shall assume R to be connected. Further, denote by ∂R the subset of function nodes that are not in F_R, but are adjacent to a variable node in V_R.

Then, for $a \in \partial R$ there exists a unique $i \in \partial a \cap V_R$, that we denote by $i(a)$. It then follows immediately from Theorem 14.1, and its characterization of messages, that the joint distribution of variables in R is

$$\mu(\underline{x}_R) = \frac{1}{Z_R} \prod_{a \in F_R} \psi_a(\underline{x}_{\partial a}) \prod_{a \in \partial R} \widehat{\nu}^*_{a \to i(a)}(x_{i(a)}) \,, \tag{14.18}$$

where $\widehat{\nu}^*_{a \to i}(\,\cdot\,)$ are the fixed point BP messages.

Exercise 14.4 Let us use the above result to write the joint distribution of variables along a path in a tree factor graph. Consider two variable nodes i, j, and let $R = (V_R, F_R, E_R)$ be the subgraph induced by nodes along the path between i and j. For any function node $a \in R$, denote by $i(a)$, $j(a)$ the variable nodes in R that are adjacent to a. Show that the joint distribution of the variables along this path, $\underline{x}_R = \{x_l : l \in V_R\}$, takes the form.

$$\mu(\underline{x}_R) = \frac{1}{Z_R} \prod_{a \in F_R} \tilde{\psi}_a(x_{i(a)}, x_{j(a)}) \prod_{l \in V_R} \tilde{\psi}_l(x_l) \,. \tag{14.19}$$

In other words $\mu(\underline{x}_R)$ factorizes according to the subgraph R. Write expressions for the compatibility functions $\tilde{\psi}_a(\,\cdot\,,\,\cdot\,)$, $\tilde{\psi}_l(\,\cdot\,)$ in terms of the original compatibility functions and of the messages going from ∂R to V_R.

A particularly useful case is the computation of the internal energy. In physics problems, the compatibility functions in Eq. (14.13) take the form $\psi_a(\underline{x}_{\partial a}) = e^{-\beta E_a(\underline{x}_{\partial a})}$, where β is the inverse temperature and $E_a(\underline{x}_{\partial a})$ is the energy function characterizing constraint a. Of course, any graphical model can be written in this form (allowing for $E_a(\underline{x}_{\partial a}) = +\infty$ in the case of hard constraints), adopting for instance the convention $\beta = 1$, that we will use hereafter. The internal energy U is the expectation value of the total energy:

$$U = -\sum_{\underline{x}} \mu(\underline{x}) \sum_{a=1}^{M} \log \psi_a(\underline{x}_{\partial a}) \,. \tag{14.20}$$

This can be computed in terms of BP messages using Eq. (14.18) with $F_R = \{a\}$. If we further use Eq. (14.14) to express products of check-to-variable messages in terms of variable-to-check ones, we get

$$U = -\sum_{a=1}^{M} \frac{1}{Z_a} \sum_{\underline{x}_{\partial a}} \left(\psi_a(\underline{x}_{\partial a}) \log \psi_a(\underline{x}_{\partial a}) \prod_{i \in \partial a} \nu^*_{i\to a}(x_j) \right) , \qquad (14.21)$$

where $Z_a \equiv \sum_{\underline{x}_{\partial a}} \psi_a(\underline{x}_{\partial a}) \prod_{i\in\partial a} \nu^*_{i\to a}(x_j)$. Notice that in this expression the internal energy is a sum of 'local' terms, one for each compatibility function.

On loopy graph Eqs. (14.18) and (14.21) are no longer valid, and indeed BP does not necessarily converge to fixed point messages $\{\nu^*_{i\to a}, \widehat{\nu}^*_{a\to i}\}$. However one can replace fixed point messages with BP messages after any number t of iterations and take these as *definitions* of the BP estimates for the corresponding quantities. From Eq. (14.18) one obtains an estimate of the joint distribution of a subset of variables, call it $\nu^{(t)}(\underline{x}_R)$, and from (14.21) an estimate of the internal energy.

14.2.4 Entropy

Remember that the entropy of a distribution μ over $\mathcal{X}^V$ is defined as $H[\mu] = -\sum_{\underline{x}} \mu(\underline{x}) \log \mu(\underline{x})$. In a tree graphical model the entropy, like the internal energy, has a simple expression in terms of local quantities. This follows from an important decomposition property. Let us denote by $\mu_a(\underline{x}_{\partial a})$ the marginal probability distribution of all the variables involved in the compatibility function a, and by $\mu_i(x_i)$ the marginal probability distribution of variable x_i.

Theorem 14.2 *In a tree graphical model, the joint probability distribution $\mu(\underline{x})$ of all the variables can be written in terms of the marginals $\mu_a(\underline{x}_{\partial a})$ and $\mu_i(x_i)$ as:*

$$\mu(\underline{x}) = \prod_{a\in F} \mu_a(\underline{x}_{\partial a}) \prod_{i\in V} \mu_i(x_i)^{1-|\partial i|} . \qquad (14.22)$$

Proof: The proof is by induction on the number M of factors. Relation (14.22) holds for $M = 1$ (since the degrees $|\partial i|$ are all equal to 1). Let us assume that it is valid for any factor graph with up to M factors, and consider a specific factor graph G with $M + 1$ factors. Since G is a tree, it contains at least one factor node such that all its adjacent variable nodes have degree 1, except at most one of them. Call such a factor node a, and let i be the only neighbor with degree larger than one (the case in which no such neighbor exists is treated analogously). Further, let $\underline{x}_\sim$ be the vector of variables in G that are not in $\partial a \setminus i$. Then (writing $\mathbb{P}_\mu(\,\cdot\,)$ for probability under the distribution μ) the Markov property together with Bayes rule yields

$$\mathbb{P}_\mu(\underline{x}) = \mathbb{P}_\mu(\underline{x}_\sim)\mathbb{P}_\mu(\underline{x}|\underline{x}_\sim) = \mathbb{P}_\mu(\underline{x}_\sim)\mathbb{P}_\mu(\underline{x}_{\partial a\setminus i}|x_i) = \mathbb{P}_\mu(\underline{x}_\sim)\mu_a(\underline{x}_{\partial a})\mu_i(x_i)^{-1} \qquad (14.23)$$

The probability $\mathbb{P}_\mu(\underline{x}_\sim)$ can be written as $\mathbb{P}(\underline{x}_\sim) \cong \tilde{\psi}_a(x_i) \prod_{b\in F\setminus a} \psi_b(\underline{x}_{\partial b})$, where $\tilde{\psi}_a(x_i) = \sum_{\underline{x}_{\partial a\setminus i}} \psi_a(\underline{x}_{\partial a})$. As the factor $\tilde{\psi}_a$ has degree one, it can be erased and incorporated in another factor as follows: take one of the other factors connected to i, $c \in \partial i \setminus a$, and change it to $\tilde{\psi}_c(\underline{x}_{\partial c}) = \psi_c(\underline{x}_{\partial c})\tilde{\psi}_a(x_i)$. In the reduced factor graph,

the degree of i is smaller by one and the number of factors is M. Using the induction hypothesis, we get

$$\mathbb{P}_\mu(\underline{x}_\sim) = \mu_i(x_i)^{2-|\partial i|} \prod_{b\in F\setminus a} \mu_b(\underline{x}_{\partial b}) \prod_{j\in V\setminus i} \mu_j(x_j)^{1-|\partial j|} \,. \tag{14.24}$$

The proof is completed by putting together Eqs. (14.23) and (14.24). □

As an immediate consequence of (14.22), the entropy of a tree graphical model can be expressed as sums of local terms:

$$H[\mu] = -\sum_{a\in F} \mu_a(\underline{x}_{\partial a}) \log \mu_a(\underline{x}_{\partial a}) - \sum_{i\in V} (1-|\partial i|)\, \mu_i(x_i) \log \mu_i(x_i) \;. \tag{14.25}$$

It is also easy to express the free-entropy $\Phi = \log Z$ in terms of *local* quantities. Recalling that $\Phi = H[\mu] - U[\mu]$ (where $U[\mu]$ is the internal energy given by Eq. (14.21)) we get $\Phi = \mathbb{F}[\mu]$, where

$$\mathbb{F}[\mu] = -\sum_{a\in F} \mu_a(\underline{x}_{\partial a}) \log\left\{\frac{\mu_a(\underline{x}_{\partial a})}{\psi_a(\underline{x}_{\partial a})}\right\} - \sum_{i\in V} (1-|\partial i|)\mu_i(x_i) \log \mu_i(x_i) \,. \tag{14.26}$$

Expressing local marginals in terms of messages, via Eq. (14.18), we can in turn write the free-entropy as a function of the fixed point messages. We shall introduce the function $\mathbb{F}_*(\underline{\nu})$, that yields the free-entropy in terms of $2|E|$ messages $\underline{\nu} = \{\nu_{i\to a}(\,\cdot\,), \widehat{\nu}_{a\to i}(\,\cdot\,)\}$:

$$\mathbb{F}_*(\underline{\nu}) = \sum_{a\in F} \mathbb{F}_a(\underline{\nu}) + \sum_{i\in V} \mathbb{F}_i(\underline{\nu}) - \sum_{(ia)\in E} \mathbb{F}_{ia}(\underline{\nu}) \tag{14.27}$$

where:

$$\mathbb{F}_a(\underline{\nu}) = \log\left[\sum_{\underline{x}_{\partial a}} \psi_a(\underline{x}_{\partial a}) \prod_{i\in\partial a} \nu_{i\to a}(x_i)\right] , \quad \mathbb{F}_i(\underline{\nu}) = \log\left[\sum_{x_i} \prod_{b\in\partial i} \widehat{\nu}_{b\to i}(x_i)\right] ,$$
$$\mathbb{F}_{ai}(\underline{\nu}) = \log\left[\sum_{x_i} \nu_{i\to a}(x_i)\widehat{\nu}_{a\to i}(x_i)\right] . \tag{14.28}$$

It is not hard to show that, evaluating this functional on the BP fixed point $\underline{\nu}^*$, one gets $\mathbb{F}_*(\underline{\nu}^*) = \mathbb{F}[\mu] = \Phi$ thus recovering the correct free-entropy. The function $\mathbb{F}_*(\underline{\nu})$ defined in (14.27) is known as the **Bethe free-entropy** (when multiplied by a factor $-1/\beta$, it is called the **Bethe free-energy**). The above observations are important enough to be highlighted in a Theorem.

Theorem 14.3. (Bethe free-entropy is exact on trees) *Consider a tree graphical model. Let $\{\mu_a, \mu_i\}$ denote its local marginals, and $\underline{\nu}^* = \{\nu^*_{i\to a}, \widehat{\nu}^*_{a\to i}\}$ be the fixed point BP messages. Then $\Phi = \log Z = \mathbb{F}[\mu] = \mathbb{F}_*(\underline{\nu}^*)$.*

Notice that in the above statement we have used the correct local marginals in $\mathbb{F}[\,\cdot\,]$ and the fixed point messages in $\mathbb{F}_*(\,\cdot\,)$. In Section 14.4 we will reconsider the Bethe free-entropy for more general graphical models, and regard it as functions over the space of all 'possible' marginals/messages.

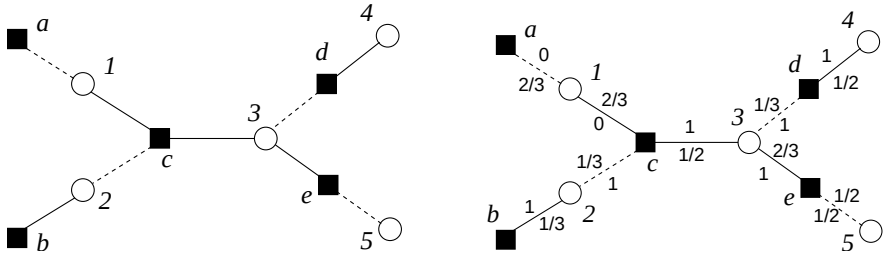

Fig. 14.4 Left: the factor graph of a small satisfiability instance with 5 variables and 5 clauses. A dashed line means that the variable appears negated in the adjacent clause. Right: the set of fixed point BP messages for the uniform measure over solutions of this instance. All messages are normalized, and we show their weight on the value "True". For any edge (a, i) (a being the clause and i the variable), the weight corresponding to the message $\widehat{\nu}_{a\to i}$ is shown above the edge, and the weight corresponding to $\nu_{i\to a}$ below the edge.

Exercise 14.5 Consider the satisfiability instance in Fig. 14.4, left. Show by exhaustive enumeration that it has only two satisfying assignments, $\underline{x} = (0, 1, 1, 1, 0)$ and $(0, 1, 1, 1, 1)$. Re-derive this result using BP. Namely, compute the entropy of the uniform measure over satisfying assignments, and check that its value is indeed $\log 2$. The BP fixed point is shown in Fig. 14.4, right.

Exercise 14.6 In many systems some of the function nodes have degree 1 and amount to a local redefinition of the reference measure over $\mathcal{X}$. It is then convenient to single out these factors. Let us write $\mu(\underline{x}) \cong \prod_{a\in F} \psi_a(\underline{x}_{\partial a}) \prod_{i\in V} \psi_i(x_i)$, where the second product runs over degree-1 function nodes (indexed by the adjacent variable node), while the factors ψ_a have degree at least 2. In the computation of $\mathbb{F}_*$, the introduction of ψ_i adds N extra factor nodes and subtracts N extra 'edge' terms corresponding to the edge between the variable node i and the function node corresponding to ψ_i. Show that these two effects cancel, and that the net effect is to replace the variable node contribution in Eq. (14.27) with

$$\mathbb{F}_i(\underline{\nu}) = \log\left[\sum_{x_i} \psi_i(x_i) \prod_{a\in\partial i} \widehat{\nu}_{a\to i}(x_i)\right] . \tag{14.29}$$

The problem of sampling from the distribution $\mu(\underline{x})$ over the large-dimensional space $\mathcal{X}^N$ reduces to the one of computing one-variable marginals of $\mu(\underline{x})$, conditional on a subset of the other variables. In other words, if we have a black box that computes $\mu(x_i|\underline{x}_U)$ for any subset $U \subseteq V$, it can be used to sample a random configuration $\underline{x}$. The standard procedure for doing this is called **sequential importance sampling**. Let us describe it in the case of tree-graphical models, using BP to implement such a 'black box':

BP-Guided Sampling (Graphical model (G, ψ))

initialize BP messages;
initialize $U = \emptyset$;
for $t = 1, \dots, N$:
    run BP until convergence;
    choose $i \in V \setminus U$;
    compute the BP marginal $\nu_i(x_i)$;
    choose x_i^* distributed according to ν_i;
    fix $x_i = x_i^*$ and set $U \leftarrow U \cup \{i\}$;
    add a factor $\mathbb{I}(x_i = x_i^*)$ to the graphical model;
end
return $\underline{x}^*$.

14.2.5 Pairwise models

Pairwise graphical models, i.e. graphical models such that all factor nodes have degree 2, form an important class. A pairwise model can be conveniently represented as an ordinary graph $G = (V, E)$ over variable nodes. An edge joins two variables each time they are the arguments of the same compatibility function. The corresponding probability distribution reads

$$\mu(\underline{x}) = \frac{1}{Z} \prod_{(ij) \in E} \psi_{ij}(x_i, x_j) \, . \tag{14.30}$$

Function nodes can be identified with edges $(ij) \in E$.

In this case belief propagation can be described as operating directly on G. Further, one of the two types of messages can be easily eliminated: we shall work uniquely with variable-to-function messages, that we will denote as $\nu_{i \to j}^{(t)}(x_i)$, a shortcut for $\nu_{i \to (ij)}^{(t)}(x_i)$. The BP updates then read

$$\nu_{i \to j}^{(t+1)}(x_i) \cong \prod_{l \in \partial i \setminus j} \sum_{x_l} \psi_{il}(x_i, x_l) \, \nu_{l \to i}^{(t)}(x_l) \, . \tag{14.31}$$

Simplified expressions can be derived in this case for the joint distribution of several variables, cf. Eq. (14.18), as well as for the free-entropy:

Exercise 14.7 Show that, for pairwise models, the free-entropy of Eq. (14.27) can be written as $\mathbb{F}_*(\underline{\nu}) = \sum_{i\in V}\mathbb{F}_i(\underline{\nu}) - \sum_{(ij)\in E}\mathbb{F}_{(ij)}(\underline{\nu})$, where:

$$\mathbb{F}_i(\underline{\nu}) = \log\left[\sum_{x_i}\prod_{j\in\partial i}\left(\sum_{x_j}\psi_{ij}(x_i,x_j)\nu_{j\to i}(x_j)\right)\right] ,$$
$$\mathbb{F}_{(ij)}(\underline{\nu}) = \log\left[\sum_{x_i,x_j}\nu_{i\to j}(x_i)\psi_{ij}(x_i,x_j)\nu_{j\to i}(x_j)\right] . \qquad (14.32)$$

14.3 Optimization: max-product and min-sum

Message passing algorithms are not limited to computing marginals. Imagine that you are given a probability distribution $\mu(\,\cdot\,)$ as in Eq. (14.13), and you are asked to find a configuration $\underline{x}$ which maximizes the probability $\mu(\underline{x})$. Such a configuration is called a **mode** of $\mu(\,\cdot\,)$. This task is important for many applications, ranging from MAP estimation (e.g. in image reconstruction) to word MAP decoding.

It is not hard to devise a message passing algorithm adapted to this task, which correctly solves the problem on trees.

14.3.1 Max-marginals

The role of marginal probabilities is here played by the so-called **max-marginals**

$$M_i(x_i^*) = \max_{\underline{x}}\{\mu(\underline{x}) : x_i = x_i^*\} . \qquad (14.33)$$

In the same way as sampling and computing partition functions can be reduced to computing marginals, optimization can be reduced to computing max-marginals. In other words, given a black box that computes max-marginals, optimization can be performed efficiently.

Consider first the simpler case in which the max-marginals are non-degenerate, i.e. for each $i \in V$, there exists x_i^* such that $M_i(x_i^*) > M_i(x_i)$ (strictly) for any $x_i \neq x_i^*$. Then the unique maximizing configuration is given by $\underline{x}^* = (x_1^*, \dots, x_N^*)$.

In the general case, the following 'decimation' procedure, which is closely related to the BP-guided sampling algorithm of Sec. 14.2.4, returns one of the maximizing configurations. Choose an ordering of the variables, say $(1, \dots, N)$. Compute $M_1(x_1)$ and let x_1^* be one of the values maximizing it: $x^* \in \arg\max M_1(x_1)$. Fix x_1 to take this value, i.e. modify the graphical model by introducing the factor $\mathbb{I}(x_1 = x_1^*)$ (this corresponds to considering the conditional distribution $\mu(\underline{x}|x_1 = x_1^*)$). Compute $M_2(x_2)$ for the new model, fix x_2 to one value $x_2^* \in \arg\max M_2(x_2)$ and iterate this procedure fixing sequentially all the x_i's.

14.3.2 Message passing

It is clear from the above that max-marginals only need to be computed up to a multiplicative normalization. We shall therefore stick to our convention of denoting by $\cong$ equality between max-marginals up to an overall normalization. Adapting the message passing update rules to the computation of max-marginals is not hard: it is sufficient to replace sums with maximizations. This yields the following **max-product** update rules:

$$\nu_{i\to a}^{(t+1)}(x_i) \cong \prod_{b\in\partial i\setminus a} \widehat{\nu}_{b\to i}^{(t)}(x_i)\,, \tag{14.34}$$

$$\widehat{\nu}_{a\to i}^{(t)}(x_i) \cong \max_{\underline{x}_{\partial a\setminus i}} \left\{ \psi_a(\underline{x}_{\partial a}) \prod_{j\in\partial a\setminus i} \nu_{j\to a}^{(t)}(x_j) \right\}\,. \tag{14.35}$$

The fixed-point conditions for this recursion are also called **max-product equations**. As in BP, it is understood that, when $\partial j\setminus a$ is an empty set, $\nu_{j\to a}(x_j)\cong 1$ is the uniform distribution. Similarly, if $\partial a\setminus j$ is empty, then $\widehat{\nu}_{a\to j}(x_j)\cong\psi_a(x_j)$. After any number of iterations, an estimate of the max-marginals is obtained as follows

$$\nu_i^{(t)}(x_i) \cong \prod_{a\in\partial i} \widehat{\nu}_{a\to i}^{(t-1)}(x_i)\,. \tag{14.36}$$

As in the case of BP, the main motivation for the above updates comes from the analysis of graphical models on trees.

Theorem 14.4. (Max-product is exact on trees) *Consider a tree graphical model with diameter t_*. Then*

1. *Irrespective of the initialization, the max-product updates (14.34), (14.35) converge after at most t_* iterations. In other words, for any edge (i,a), and any $t>t_*$ $\nu_{i\to a}^{(t)}=\nu_{i\to a}^{*}$, $\widehat{\nu}_{a\to i}^{(t)}=\widehat{\nu}_{a\to i}^{*}$.*
2. *The max-marginals are estimated correctly, i.e. for any variable node i, and ant $t>t_*$, $\nu_i^{(t)}(x_i)=M_i(x_i)$.*

The proof follows closely the one of Theorem 14.1, and is left as an exercise for the reader.

Exercise 14.8 The crucial property used both in both Theorems 14.1 and 14.4 is the distributive property of sum and max with respect to the product. Consider for instance a function of the form $f(x_1, x_2, x_3) = \psi_1(x_1, x_2)\psi_2(x_1, x_3)$. Then one can decompose the sum and max as

$$\sum_{x_1,x_2,x_3} f(x_1, x_2, x_3) = \sum_{x_1} \left[\left(\sum_{x_2} \psi_1(x_1, x_2) \right) \left(\sum_{x_3} \psi_2(x_1, x_3) \right) \right] , \tag{14.37}$$

$$\max_{x_1,x_2,x_3} f(x_1, x_2, x_3) = \max_{x_1} \left[\left(\max_{x_2} \psi_1(x_1, x_2) \right) \left(\max_{x_3} \psi_2(x_1, x_3) \right) \right] . \tag{14.38}$$

Formulate a general 'marginalization' problem (with the ordinary sum and product substituted by general operations with a distributive property) and describe a message passing algorithm that solves it on trees.

The max-product messages $\nu_{i\to a}^{(t)}(\,\cdot\,)$, $\widehat{\nu}_{a\to i}^{(t)}(\,\cdot\,)$ admit an interpretation which is analogous to the one of pum-product messages. For instance $\nu_{i\to a}^{(t)}(\,\cdot\,)$ is an estimate of the max-marginal of variable x_i with respect to the modified graphical model in which factor node a is removed from the graph. Along with the proof of Theorem 14.4, it is easy to show that, on a tree-graphical model, fixed point messages do indeed coincide with max-marginals of such modified graphical models.

The problem of finding the mode of a distribution that factorizes as in Eq. (14.13) has an alternative formulation, namely minimizing a cost (energy) function that can be written as the sum of local terms:

$$E(\underline{x}) = \sum_{a\in F} E_a(\underline{x}_{\partial a}) \, . \tag{14.39}$$

The problems are mapped onto each other by writing $\psi_a(\underline{x}_{\partial a}) = e^{-\beta E_a(\underline{x}_{\partial a})}$ (with β some positive constant). A set of message passing rules that is better adapted to the last formulation is obtained by taking the logarithm of Eqs. (14.34), (14.35). This version of the algorithm is known as **min-sum**:

$$E_{i\to a}^{(t+1)}(x_i) = \sum_{b\in\partial i\setminus a} \widehat{E}_{b\to i}^{(t)}(x_i) + C_{i\to a}^{(t)} \, , \tag{14.40}$$

$$\widehat{E}_{a\to i}^{(t)}(x_i) = \min_{\underline{x}_{\partial a\setminus i}} \left[E_a(\underline{x}_{\partial a}) + \sum_{j\in\partial a\setminus i} E_{j\to a}^{(t)}(x_j) \right] + \widehat{C}_{a\to i}^{(t)} \, . \tag{14.41}$$

The corresponding fixed-point equations are also known in statistical physics as the **energetic cavity equations**. Notice that, since the max-product marginals are relevant up to a multiplicative constant, the min-sum messages are defined up to an overall additive constant. In the following we will choose the constant $C_{i\to a}^{(t)}$ (respectively $\widehat{C}_{a\to i}^{(t)}$) such that $\min_{x_i} E_{i\to a}^{(t+1)}(x_i) = 0$ (respectively $\min_{x_i} \widehat{E}_{a\to i}^{(t)}(x_i) = 0$). The

analogous of the max-marginal estimate in Eq. (14.36) is provided by the following log-max-marginal

$$E_i^{(t)}(x_i) = \sum_{a \in \partial i} \widehat{E}_{a\to i}^{(t-1)}(x_i) + C_i^{(t)} \, . \tag{14.42}$$

In the case of tree graphical models, the minimum energy $U_* = \min_{\underline{x}} E(\underline{x})$ can be immediately written in terms of the fixed point messages $\{E^*_{i\to a}, \widehat{E}^*_{i\to a}\}$. We get indeed

$$U_* = \sum_a E_a(\underline{x}^*_{\partial a}) \, , \tag{14.43}$$

$$\underline{x}^*_{\partial a} = \arg\min_{\underline{x}_{\partial a}} \left\{ E_a(\underline{x}_{\partial a}) + \sum_{i \in \partial a} \widehat{E}^*_{i\to a}(x_i) \right\} . \tag{14.44}$$

In the case of non-tree graphs, this can be taken as a prescription to obtain a max-product estimate $U_*^{(t)}$ of the minimum energy. One just needs to replace the fixed point messages in Eq. (14.44) with the ones obtained after t iterations. Finally, a minimizing configuration $\underline{x}^*$ can be obtained through the decimation procedure described in the previous section.

Exercise 14.9 Show that U_* is also given by $U_* = \sum_{a\in F} \epsilon_a + \sum_{i\in V} \epsilon_i - \sum_{(ia)\in E} \epsilon_{ia}$, where:

$$\epsilon_a = \min_{\underline{x}_{\partial a}} \left[E_a(\underline{x}_{\partial a}) + \sum_{j\in\partial a} E^*_{j\to a}(x_j) \right] , \qquad \epsilon_i = \min_{x_i} \left[\sum_{a\in\partial i} \widehat{E}^*_{a\to i}(x_i) \right] ,$$
$$\epsilon_{ia} = \min_{x_i} \left[E^*_{i\to a}(x_i) + \widehat{E}^*_{a\to i}(x_i) \right] . \tag{14.45}$$

[Hints: (*i*) Define $x_i^*(a) = \arg\min \left[\widehat{E}^*_{a\to i}(x_i) + E^*_{i\to a}(x_i) \right]$, and show that the minima in Eqs. (14.45) are achieved at $x_i = x_i^*(a)$ (for ϵ_i and ϵ_{ai}), and at $\underline{x}^*_{\partial a} = \{x_i^*(a)\}_{i\in\partial a}$ (for ϵ_a); (*ii*) Show that $\sum_{(ia)} \widehat{E}^*_{a\to i}(x_i^*(a)) = \sum_i \epsilon_i$.]

14.3.3 Warning propagation

A frequently encountered case is that of constraint satisfaction problems, where the energy function just counts twice the number of violated constraints:

$$E_a(\underline{x}_{\partial a}) = \begin{cases} 0 & \text{if constraint } a \text{ is satisfied,} \\ 1 & \text{otherwise.} \end{cases} \tag{14.46}$$

The structure of messages can be simplified considerably in this case. More precisely, if the messages are initialized in such a way that $\widehat{E}^{(0)}_{a\to i} \in \{0, 1\}$, this condition is preserved by the min-sum updates (14.41), (14.40) at any subsequent time. Let us

prove this statement by induction. Suppose it holds up to time $t-1$. From Eq. (14.41) it follows that $E^{(t)}_{i\to a}(x_i)$ is a non-negative integer. Consider now Eq. (14.40). Since both $E^{(t)}_{j\to a}(x_j)$ and $E_a(\underline{x}_{\partial a})$ are integers, $\widehat{E}^{(t)}_{a\to i}(x_i)$, the minimum of the right hand side is a non-negative integer as well. Further, since for each $j\in\partial a\setminus i$ there exists x^*_j such that $E^{(t)}_{j\to a}(x^*_j)=0$, the minimum in Eq. (14.40) is at most 1, which proves our claim.

This argument also shows that the outcome of the minimization in Eq. (14.40) only depends on which entries of the messages $E^{(t)}_{j\to a}(\,\cdot\,)$ are vanishing. If there exists an assignment x^*_j, such that $E^{(t)}_{j\to a}(x^*_j)=0$ for each $j\in\partial a\setminus i$, and $E_a(x_i,\underline{x}^*_{\partial a\setminus i})=0$, then the value of the minimum is 0. Otherwise it is 1.

In other words, instead of keeping track of the messages $E_{i\to a}(\,\cdot\,)$, one can use their 'projections'

$$\mathtt{E}_{i\to a}(x_i)=\min\{1,\,E_{i\to a}(x_i)\}\ . \tag{14.47}$$

Proposition 14.5 *Consider an optimization problem with cost function of the form (14.39) with $E_a(\underline{x}_{\partial a})\in\{0,1\}$, and assume the min-sum algorithm to be initialized with $\widehat{E}_{a\to i}(x_i)\in\{0,1\}$ for all edges (i,a). Then, after any number of iterations, the function node-to-variable node messages coincide with the ones computed with the following update rules*

$$\mathtt{E}^{(t+1)}_{i\to a}(x_i)=\min\left\{1,\,\sum_{b\in\partial i\setminus a}\widehat{E}^{(t)}_{b\to i}(x_i)+C^{(t)}_{i\to a}\right\}\ , \tag{14.48}$$

$$\widehat{E}^{(t)}_{a\to i}(x_i)=\min_{\underline{x}_{\partial a\setminus i}}\left\{E_a(\underline{x}_{\partial a})+\sum_{j\in\partial a\setminus i}\mathtt{E}^{(t)}_{j\to a}(x_j)\right\}+\widehat{C}^{(t)}_{a\to i}\ , \tag{14.49}$$

where $C^{(t)}_{i\to a}$, $\widehat{C}^{(t)}_{a\to i}$ are normalization constants determined by $\min_{x_i}\widehat{E}_{a\to i}(x_i)=0$ and $\min_{x_i}\mathtt{E}_{i\to a}(x_i)=0$.

Finally, the ground state energy takes the same form as (14.45), with $\mathtt{E}_{i\to a}(\,\cdot\,)$ replacing $E_{i\to a}(\,\cdot\,)$.

We shall call **warning propagation** the simplified min-sum algorithm with update equations (14.49), (14.48).

The name is due to the remark that the messages $\mathtt{E}_{i\to a}(\,\cdot\,)$ can be interpreted as the following warnings:

$\mathtt{E}_{i\to a}(x_i)=1\ \to$ *"according to the set of constraints $b\in\partial i\setminus a$, the i-th variable should not take value x_i."*

$\mathtt{E}_{i\to a}(x_i)=0\to$ *"according to the set of constraints $b\in\partial i\setminus a$, the i-th variable can take value x_i."*

Warning propagation provides a procedure for finding all direct implications of some partial assignment of the variables in a constraint satisfaction problem. For instance, in satisfiability it finds all implications found by unit clause propagation, cf. Sec. 10.2.

14.4 Loopy BP

We have seen how message passing algorithms can be used efficiently on tree-graphical models. In particular they allow to exactly sample, compute marginals, partition functions, modes of distributions that factorize according to tree factor graphs. It would be very important for a number of applications to accomplish the same tasks when the underlying factor graph is no longer a tree.

It is tempting to use the BP equations in this more general context, hoping to get approximate results for large graphical models. Often we shall be dealing with problems that are NP-hard, even to approximate, and it is difficult to provide general guarantees of performance. Indeed, an important unsolved challenge is to identify classes of graphical models where the following questions can be answered:

1. Is there any set of messages $\{\nu^*_{i\to a}, \widehat{\nu}^*_{a\to i}\}$ that reproduces the local marginals of $\mu(\,\cdot\,)$ through Eq. (14.18), within some prescribed accuracy?

2. Do such messages correspond to an (approximate) fixed point of the BP update rules (14.14), (14.15)?

3. Do the BP update rules have at least one (approximate) fixed point? Is it unique?

4. Does such a fixed point have a non-empty 'basin of attraction' with respect to Eqs. (14.14), (14.15)? Does this basin of attraction include all possible (or all 'reasonable') initializations?

We shall not treat these questions in depth, as a general theory is lacking. We shall rather describe the rather sophisticated picture that has emerged, building on a mixture of physical intuition and methods, empirical observations, and rigorous proofs.

Exercise 14.10 Consider a ferromagnetic Ising model on the two dimensional grid with periodic boundary conditions (i.e. 'wrapped' on a torus), defined in Sec. 9.1.2, cf. Fig. 9.7. Ising spins σ_i, $i \in V$ are associated to the vertices of the grid, and interact along the edges:

$$\mu(\underline{\sigma}) = \frac{1}{Z}\, e^{\beta \sum_{(ij)\in E} \sigma_i \sigma_j}\,. \tag{14.50}$$

(*a*) Describe the associated factor graph.

(*b*) Write the BP equations.

(*c*) Look for a solution that is invariant under translation $\nu_{i\to a}(\sigma_i) = \nu(\sigma_i)$, $\widehat{\nu}_{a\to i}(\sigma_i) = \widehat{\nu}(\sigma_i)$: write the equations satisfied by $\nu(\,\cdot\,)$, $\widehat{\nu}(\,\cdot\,)$.

(*d*) Parameterize $\nu(\sigma)$ in terms of the log-likelihood $h = \frac{1}{2\beta}\log\frac{\nu(+1)}{\nu(-1)}$ and show that h satisfies the equation $\tanh(\beta h) = \tanh(\beta)\tanh(3\beta h)$.

(*e*) Study this equation and show that, for $3\tanh\beta > 1$, it has three distinct solutions corresponding to three BP fixed points.

(*f*) Consider iterating the BP updates starting from a translation invariant initial condition. Does the iteration converge to a fixed point? Which one?

(*g*) Discuss the appearance of three BP fixed points in relation with the structure of the distribution $\mu(\underline{\sigma})$, and the paramagnetic-ferromagnetic transition. What is the approximate value of the critical temperature obtained from BP? Compare with the exact value $\beta_{\rm c} = \frac{1}{2}\log(1+\sqrt{2})$.

(*h*) What results does one obtain for an Ising model on a d-dimensional (instead of two-dimensional) grid?

14.4.1 Bethe free-entropy and variational methods

As we saw in Section 14.2.4, the free-entropy of a tree graphical model has a simple expression in terms of local marginals, cf. Eq. (14.26). We can use it in graphs with loops with the hope that it provides a good estimate of the actual free-entropy. In spirit this approach is similar to the 'mean field' free-entropy introduced in Ch. 2, although it differs from it in several respects.

In order to define precisely the Bethe free-entropy, we must first describe a space of 'possible' local marginals. A minimalistic approach is to restrict ourselves to the so-called 'locally consistent marginals'. A set of **locally consistent marginals** is a collection of distributions $b_i(\,\cdot\,)$ over $\mathcal{X}$, for each $i \in V$, and $b_a(\,\cdot\,)$ over $\mathcal{X}^{|\partial a|}$ for each $a \in F$. Being distributions they must be non-negative, $b_i(x_i) \geq 0$ $b_a(\underline{x}_{\partial a}) \geq 0$, and they must satisfy the normalization condition

$$\sum_{x_i} b_i(x_i) = 1 \quad \forall i \in V\,, \qquad \sum_{\underline{x}_{\partial a}} b_a(\underline{x}_{\partial a}) = 1 \quad \forall a \in F\,. \tag{14.51}$$

To be 'locally consistent', they must satisfy the marginalization condition:

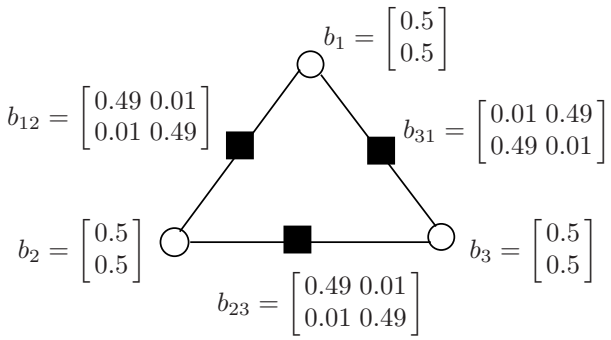

Fig. 14.5 A set of locally consistent marginals, 'beliefs', that cannot arise as the marginals of any global distribution.

$$\sum_{\underline{x}_{\partial a \setminus i}} b_a(\underline{x}_{\partial a}) = b_i(x_i) \qquad \forall a \in F\,, \;\; \forall i \in \partial a\,. \tag{14.52}$$

Given a factor graph G, we shall denote the set locally consistent marginals as $\mathsf{LOC}(G)$, and the Bethe free-entropy will be defined as a real valued function on this space.

It is important to stress that, although the marginals of any probability distribution $\mu(\underline{x})$ over $\underline{x} = (x_1, \dots, x_N)$ must be locally consistent, the converse is not true: one can find sets of locally consistent marginals that do not correspond to any distribution. In order to emphasize this point, locally consistent marginals are sometimes called "**beliefs**".

Exercise 14.11 Consider the graphical model in Fig. 14.4.1, on binary variables (x_1, x_2, x_3), $x_i \in \{0, 1\}$. The figure also gives a set of beliefs in the vector/matrix form:

$$b_i = \begin{bmatrix} b_i(0) \\ b_i(1) \end{bmatrix} \;\; ; \;\; b_{ij} = \begin{bmatrix} b_{ij}(00) & b_{ij}(01) \\ b_{ij}(10) & b_{ij}(11) \end{bmatrix}. \tag{14.53}$$

Check that this set of beliefs is locally consistent, but they cannot be the marginals of any distribution $\mu(x_1, x_2, x_3)$.

Given a set of locally consistent marginals $\underline{b} = \{b_a, b_i\}$, we associate to it a **Bethe free-entropy** exactly as in Eq. (14.26)

$$\mathbb{F}[\underline{b}] = -\sum_{a \in F} b_a(\underline{x}_{\partial a}) \log \left\{ \frac{b_a(\underline{x}_{\partial a})}{\psi_a(\underline{x}_{\partial a})} \right\} - \sum_{i \in V} (1 - |\partial i|)\, b_i(x_i) \log b_i(x_i)\,. \tag{14.54}$$

The analogy with naive mean field suggests that stationary points (and in particular maxima) of the Bethe free-entropy should play an important role. This is partially confirmed by the following result.

Proposition 14.6 *Assume $\psi_a(\underline{x}_{\partial a}) > 0$ for each a and $\underline{x}_{\partial a}$. Then the stationary points of the Bethe free-entropy $\mathbb{F}[\underline{b}]$ are in one-to-one correspondence with the fixed points of BP.*

As it will appear from the proof, the correspondence between BP fixed points and stationary points of $\mathbb{F}[\underline{b}]$ is completely explicit.

Proof: We want to check stationarity with respect to variations of $\underline{b}$ within the set $\mathsf{LOC}(G)$, that is defined by the constraints (14.51), (14.52), as well as $b_a(\underline{x}_{\partial a}) \geq 0$, $b_i(x_i) \geq 0$. We thus introduce a set of Lagrange multipliers $\underline{\lambda} = \{\lambda_i,\, i \in V;\ \lambda_{ai}(x_i),\, (a,i) \in E,\, x_i \in \mathcal{X}\}$, where λ_i corresponds to the normalization of $b_i(\,\cdot\,)$ and $\lambda_{ai}(x_i)$ corresponds to the marginal of b_a coinciding with b_i. We then define the Lagrangian

$$\mathcal{L}(\underline{b},\underline{\lambda}) = \mathbb{F}[\underline{b}] - \sum_{a\in F}\lambda_i\left[\sum_{x_i} b_i(x_i) - 1\right] - \sum_{(ia),x_i}\lambda_{ai}(x_i)\left[\sum_{\underline{x}_{\partial a\setminus i}} b_a(\underline{x}_{\partial a}) - b_i(x_i)\right] . \tag{14.55}$$

Notice that we did not introduce a Lagrange multiplier for the normalization of $b_a(\underline{x}_{\partial a})$ as this follows from the two constraints already enforced. The stationarity conditions with respect to b_i and b_a imply:

$$b_i(x_i) \cong e^{-\frac{1}{|\partial i|-1}\sum_{a\in\partial i}\lambda_{ai}(x_i)} , \quad b_a(\underline{x}_{\partial a}) \cong \psi_a(\underline{x}_{\partial a})\, e^{-\sum_{i\in\partial a}\lambda_{ai}(x_i)} . \tag{14.56}$$

The Lagrange multipliers must be chosen in such a way that Eq. (14.52) is fulfilled. Any such set of Lagrange multipliers yields a stationary point of $\mathbb{F}[\underline{b}]$. Once the $\lambda_{ai}(x_j)$ are found, the computation of the normalization constants in these expressions fixes λ_i. Conversely, any stationary point corresponds to a set of Lagrange multipliers satisfying the stated condition.

It remains to show that sets of Lagrange multipliers such that $\sum_{\underline{x}_{\partial a\setminus i}} b_a(\underline{x}_{\partial a}) = b_i(x_i)$ are in one-to-one correspondence with BP fixed points. In order to see this, define the messages

$$\nu_{i\to a}(x_i) \cong e^{-\lambda_{ai}(x_i)} , \quad \widehat{\nu}_{a\to i}(x_i) \cong \sum_{\underline{x}_{\partial a\setminus i}} \psi_a(\underline{x}_{\partial a})\, e^{-\sum_{j\in\partial a\setminus i}\lambda_{aj}(x_j)} . \tag{14.57}$$

It is clear from the definition that such messages satisfy

$$\widehat{\nu}_{a\to i}(x_i) \cong \sum_{\underline{x}_{\partial a\setminus i}} \psi_a(\underline{x}_{\partial a}) \prod_{j\in\partial a\setminus i} \nu_{i\to a}(x_i) . \tag{14.58}$$

Further, using the second of Eqs. (14.56) together with (14.57) we get $\sum_{\underline{x}_{\partial a\setminus i}} b_a(\underline{x}_{\partial a}) \cong \nu_{i\to a}(x_i)\widehat{\nu}_{a\to i}(x_i)$. On the other hand, from the first of Eqs. (14.56) together with (14.57), we get $b_i(x_i) \cong \prod_b \nu_{i\to b}(x_i)^{\frac{1}{|\partial i|-1}}$. The marginalization condition thus implies

$$\prod_{b\in\partial i} \nu_{i\to b}(x_i)^{\frac{1}{|\partial i|-1}} \cong \nu_{i\to a}(x_i)\widehat{\nu}_{a\to i}(x_i) . \tag{14.59}$$

Taking the product of these equalities for $a \in \partial i \setminus b$, and eliminating $\prod_{a\in\partial i\setminus b}\nu_{i\to a}(x_i)$ from the resulting equation (which is possible if $\psi_a(\underline{x}_{\partial a}) > 0$), we get

$$\nu_{i\to b}(x_i) \cong \prod_{a\in\partial i\setminus b} \widehat{\nu}_{a\to i}(x_i) . \tag{14.60}$$

At this point we recognize in Eqs. (14.58), (14.60) the fixed point condition for BP, cf. Eqs. (14.14), (14.15). Conversely, given any solution of Eqs. (14.58), (14.60) one

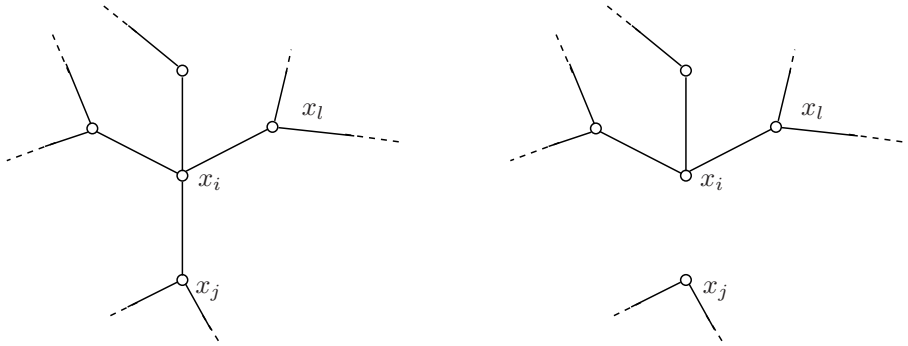

Fig. 14.6 Neighborhood of node i in a pairwise graphical model. Right: the modified graphical model used to define message $\nu_{i\to j}(x_i)$.

can define a set of Lagrange multipliers using the first of Eqs. (14.57). It follows from the fixed point condition that the second Eq. (14.57) is fulfilled as well, and that the marginalization condition holds. □

An important consequence of this proposition is the existence of BP fixed points.

Corollary 14.7 *Assume $\psi_a(\underline{x}_a) > 0$ for each a and $\underline{x}_{\partial a}$. Then BP has at least one fixed point.*

Proof: Since $\mathbb{F}[\underline{b}]$ is bounded and continuous in $\mathsf{LOC}(G)$ (which is closed), it takes its maximum at some point $\underline{b}^* \in \mathsf{LOC}(G)$. Using the condition $\psi_a(\underline{x}_a) > 0$ it is easy to see that such a maximum is reached in the relative interior of $\mathsf{LOC}(G)$, i.e. that $b^*_a(\underline{x}_{\partial a}) > 0$, $b^*_i(x_i) > 0$ strictly. As a consequence $\underline{b}^*$ must be a stationary point and therefore, by Proposition 14.6, there is a BP fixed point associated with it. □

The 'variational principle' provided by Proposition 14.6 is particularly suggestive as it is analogous to naive mean field bounds. For practical applications it is sometimes more convenient to use the free-entropy functional $\mathbb{F}_*(\underline{\nu})$ of Eq.(14.27). This can be regarded as a function from the space of messages to reals: $\mathbb{F} : \mathfrak{M}(\mathcal{X})^{|\vec{E}|} \to \mathbb{R}$ (remember that $\mathfrak{M}(\mathcal{X})$ denotes the set of measures over $\mathcal{X}$, and $\vec{E}$ is the set of directed edges in the factor graph)[3]. It satisfies the following variational principle.

Proposition 14.8 *The stationary points of the Bethe free-entropy $\mathbb{F}_*(\underline{\nu})$ are fixed points of belief propagation. Conversely, any fixed point $\underline{\nu}$ of belief propagation such that $\mathbb{F}_*(\underline{\nu})$ is finite, is also a stationary point of $\mathbb{F}_*(\underline{\nu})$.*

The proof is simple calculus and is left to the reader.

It turns out that for tree graphs and for unicyclic graphs, $\mathbb{F}[\underline{b}]$ is convex, and the above results then prove the existence and unicity of BP fixed points. But for general graphs $\mathbb{F}[\underline{b}]$ is non-convex and may have multiple stationary points.

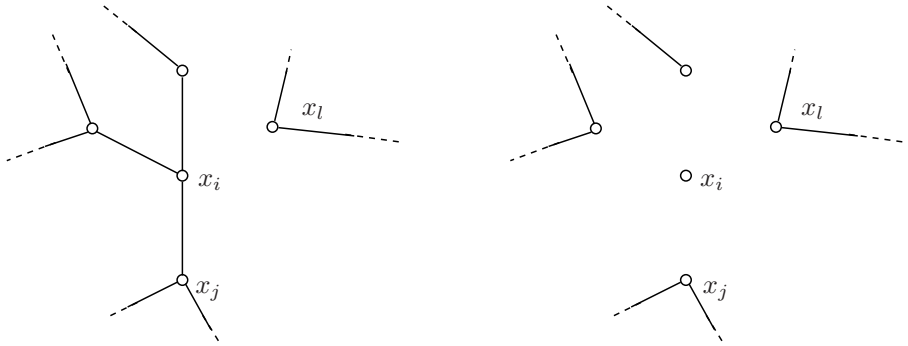

Fig. 14.7 Left: Modified graphical model used to define $\nu_{l\to i}(x_l)$. Right: Modified graphical model corresponding to the cavity distribution of the neighbors of i, $\mu_{\partial i\setminus j}(\underline{x}_{\partial i\setminus j})$.

14.4.2 Correlations

What is the origin of the error made when using BP in an arbitrary graph with loops, and under what conditions can it be small? In order to understand this point, let us consider for notational simplicity a pairwise graphical model, cf. Eq. (14.2.5). The generalization to other models is straightforward. Taking seriously the probabilistic interpretation of messages, we want to compute the marginal distribution $\nu_{i\to j}(x_i)$ of x_i in the modified graphical model that does not include the factor $\psi_{ij}(x_i, x_j)$ (see Fig. 14.6). Call $\mu_{\partial i\setminus j}(\underline{x}_{\partial i\setminus j})$ the joint distribution of all variables in $\partial i\setminus j$ in the model where all the factors $\psi_{il}(x_i, x_l)$, $l\in\partial i$, have been removed. Then:

$$\nu_{i\to j}(x_i) \cong \sum_{\underline{x}_{\partial i\setminus j}} \prod_{l\in\partial i\setminus j} \psi_{il}(x_i, x_l)\mu_{\partial i\setminus j}(\underline{x}_{\partial i\setminus j})\,. \tag{14.61}$$

Comparing this expression to the BP equations, cf. Eq. (14.31), we deduce that the messages $\{\nu_{i\to j}\}$ solve these equations if

$$\mu_{\partial i\setminus j}(\underline{x}_{\partial i\setminus j}) = \prod_{l\in\partial i\setminus j} \nu_{l\to i}(x_l)\,. \tag{14.62}$$

We can think that this happens when two conditions are fulfilled:

1. Under $\mu_{\partial i\setminus j}(\,\cdot\,)$, the variables $\{x_l : l\in\partial i\setminus j\}$ are independent: $\mu_{\partial i\setminus j}(\underline{x}_{\partial i\setminus j}) = \prod_{l\in\partial i\setminus j}\mu_{\partial i\setminus j}(x_l)$.
2. The marginal of each of these variables under $\mu_{\partial i\setminus j}(\,\cdot\,)$ is equal to the corresponding message $\nu_{l\to i}(x_l)$. In other words the two graphical models obtained by removing all the compatibility functions that involve x_i (namely, the model $\mu_{\partial i\setminus j}(\,\cdot\,)$) and by removing only $\psi_{il}(x_i, x_l)$, must have the same marginal for variable x_l, cf. Fig. 14.7.

These two conditions are obviously fulfilled for tree graphical models. They are also approximately fulfilled if correlations among variables $\{x_l : l\in\partial i\}$ are 'small' under

[3] On a tree $\mathbb{F}_*(\underline{\nu})$ is (up to a change of variables) the Lagrangian dual of $\mathbb{F}(\underline{b})$.

$\mu_{\partial i\setminus j}(\,\cdot\,)$. As we have seen, in many cases of practical interest (LDPC codes, random K-SAT, etc.) the factor graph is locally tree-like. In other words, when removing node i, the variables $\{x_l : l \in \partial i\}$ are with high probability far apart from each other. This suggests that, in such models, the two conditions above may indeed hold in the large size limit, provided far apart variables are weakly correlated. A simple illustration of this phenomenon is provided in the exercises below. The following chapters will investigate this property further and discuss how to cope with cases in which it does not hold.

Exercise 14.12 Consider the antiferromagnetic Ising model on a ring, with variables $(\sigma_1, \dots, \sigma_N) \equiv \underline{\sigma}$, $\sigma_i \in \{+1, -1\}$ and distribution

$$\mu(\underline{\sigma}) = \frac{1}{Z}\, e^{-\beta \sum_{i=1}^{N} \sigma_i \sigma_{i+1}} \tag{14.63}$$

where $\sigma_{N+1} \equiv \sigma_1$. This is a pairwise graphical model whose graph G is the ring over N vertices.

(*a*) Write the BP update rules for this model (see Section 14.2.5).

(*b*) Express the update rules in terms of log-likelihoods $h^{(t)}_{i\to} \equiv \frac{1}{2}\log\frac{\nu^{(t)}_{i\to i+1}(+1)}{\nu^{(t)}_{i\to i+1}(-1)}$, and $h^{(t)}_{\leftarrow i} \equiv \frac{1}{2}\log\frac{\nu^{(t)}_{i\to i-1}(+1)}{\nu^{(t)}_{i\to i-1}(-1)}$.

(*c*) Show that, for any $\beta \in [0, \infty)$, and any initialization, the BP updates converge to the unique fixed point $h_{\leftarrow i} = h_{i\to} = 0$ for all i.

(*d*) Assume $\beta = +\infty$ and N even. Show that any set of log-likelihoods of the form $h_{i\to} = (-1)^i a$, $h_{\leftarrow i} = (-1)^i b$, with $a, b \in [-1, 1]$, is a fixed point.

(*e*) Consider now $\beta = \infty$ and N odd, and show that the only fixed point is $h_{\leftarrow i} = h_{i\to} = 0$. Find an initialization of the messages such that BP does not converge to this fixed point.

Exercise 14.13 Consider the ferromagnetic Ising model on a ring with magnetic field. This is defined through the distribution

$$\mu(\underline{\sigma}) = \frac{1}{Z}\, e^{\beta \sum_{i=1}^{N} \sigma_i \sigma_{i+1} + B \sum_{i=1}^{N} \sigma_i} \tag{14.64}$$

where $\sigma_{N+1} \equiv \sigma_1$. Notice that with respect to the previous exercise we changed a sign in the exponent.

(a, b) As in the previous exercise.

(c) Show that, for any $\beta \in [0, \infty)$, and any initialization, the BP updates converge to the unique fixed point $h_{\leftarrow i} = h_{i\rightarrow} = h_*(\beta, B)$ for all i.

(d) Let $\langle \sigma_i \rangle$ be the expectation of spin σ_i with respect to the measure $\mu(\,\cdot\,)$, and $\langle \sigma_i \rangle_{\rm BP}$ the corresponding BP estimate. Show that $|\langle \sigma_i \rangle - \langle \sigma_i \rangle_{\rm BP}| = O(\lambda^N)$ for some $\lambda \in (0, 1)$.

14.5 General message passing algorithms

Both the sum-product and max-product (or min-sum) algorithms are instances of a more general class of **message passing** algorithms. All the algorithms in this family share some common features that we now highlight.

Given a factor graph, a message-passing algorithm is defined by the following ingredients:

1. An alphabet of messages M. This can be either continuous or discrete. The algorithm operates on messages $\nu_{i\to a}^{(t)}, \widehat{\nu}_{a\to i}^{(t)} \in \mathsf{M}$ associated with the directed edges in the factor graph.
2. Update functions $\Psi_{i\to a} : \mathsf{M}^{|\partial i\setminus a|} \to \mathsf{M}$ and $\Phi_{a\to i} : \mathsf{M}^{|\partial a\setminus i|} \to \mathsf{M}$ that describe how to update messages.
3. An initialization, i.e. a mapping from the directed edges in the factor graph to M (it can be a random mapping). We shall denote by $\nu_{i\to a}^{(0)}$, $\widehat{\nu}_{a\to i}^{(0)}$ the image of such a mapping.
4. A decision rule, i.e. a local function from messages to a space of 'decisions' among which we are interested to make a choice. Since we will be mostly interested in computing marginals (or max-marginals), we shall assume the decision rule to be given by a family of functions $\widehat{\Psi}_i : \mathsf{M}^{|\partial i|} \to \mathfrak{M}(\mathcal{X})$.

Notice the characterizing feature of message passing algorithms: messages outgoing from a node are functions of messages incoming on the same node through the other edges.

Given these ingredients, a message passing algorithm with parallel updating is defined as follows. Assign the values of initial messages $\nu_{i\to a}^{(0)}$, $\widehat{\nu}_{a\to i}^{(0)}$ according to the initialization rule. Then, for any $t \geq 0$, update messages through local operations at variable/check nodes as follows:

$$\nu_{i\to a}^{(t+1)} = \Psi_{i\to a}(\{\widehat{\nu}_{b\to i}^{(t)} : b \in \partial i \setminus a\}) , \tag{14.65}$$

$$\widehat{\nu}_{a\to i}^{(t)} = \Phi_{a\to i}(\{\nu_{j\to a}^{(t)} : j \in \partial a \setminus i\}) . \tag{14.66}$$

Finally, after a pre-established number of iterations t, take the decision using the rules $\widehat{\Psi}_i$, namely return

$$\nu_i^{(t)}(x_i) = \widehat{\Psi}_i(\{\widehat{\nu}_{b\to i}^{(t-1)} : b \in \partial i\})(x_i) . \tag{14.67}$$

Many variants are possible concerning the update schedule. For instance in sequential updating one can pick up a directed edge uniformly at random and compute the corresponding message. Another possibility is to generate a random permutation of the edges and update the messages according to this permutation. We shall not discuss these 'details', but the reader should be aware that they can be important in practice: some update schemes may converge better than others.

Exercise 14.14 Recast the sum-product and min-sum algorithms in the general message passing framework. In particular, specify the messages alphabet, the update and decision rules.

14.6 Probabilistic analysis

In the following chapters we shall repeatedly be concerned with the analysis of message passing algorithms on random graphical models. In this context messages become random variables, and their distribution can be characterized in the large system limit, as we will now see.

14.6.1 Assumptions

Before proceeding, it is necessary to formulate a few technical assumptions under which the approach works. The basic idea is that, in a 'random graphical model', distinct nodes should be essentially independent. Specifically, we shall consider below a setting which already includes many cases of interest; it is easy to extend our analysis to even more general situations.

A **random graphical model** is a (random) probability distribution on $\underline{x} = (x_1, \dots, x_N)$ of the form[4]

$$\mu(\underline{x}) \cong \prod_{a\in F} \psi_a(\underline{x}_{\partial a}) \prod_{i\in V} \psi_i(x_i) , \tag{14.68}$$

where the factor graph $G = (V, F, E)$ (with variable nodes V, factor nodes F, and edges E), and the various factors ψ_a, ψ_i, are independent random variables. More precisely, we assume that the factor graph is distributed according to one of the ensembles $\mathbb{G}_N(K, \alpha)$ or $\mathbb{D}_N(\Lambda, P)$ (see Ch. 9).

[4]Notice that the factors ψ_i, $i \in V$ could have been included as degree 1 function nodes as we do in (14.13); including them explicitly yields a description of density evolution which is more symmetric between variables and factors, and applies more directly to decoding

The random factors are assumed to be distributed as follows. For any given degree k, we are given a list of possible factors $\psi^{(k)}(x_1,\dots,x_k;\widehat{J})$, indexed by a 'label' $\widehat{J}\in\mathsf{J}$, and a distribution $P^{(k)}_{\widehat{J}}$ over the set of possible labels J. For each function node $a\in F$ of degrees $|\partial a|=k$, a label $\widehat{J}_a$ is drawn with distribution $P^{(k)}_{\widehat{J}}$, and the function $\psi_a(\,\cdot\,)$ is taken equal to $\psi^{(k)}(\,\cdot\,;\widehat{J}_a)$. Analogously, the factors ψ_i are drawn from a list of possible $\{\psi(\,\cdot\,;J)\}$, indexed by the label J which is drawn from a distribution P_J. The random graphical model is fully characterized by the graph ensemble, the set of distributions $P^{(k)}_{\widehat{J}}$, P_J, and the lists of factors $\{\psi^{(k)}(\,\cdot\,;\widehat{J})\}$, $\{\psi(\,\cdot\,;J)\}$.

We need to make some assumptions on the message update rules. Specifically, we assume that the variable-to-function node update rules $\Psi_{i\to a}$ depend on $i\to a$ only through $|\partial i|$ and J_i, and the function-to-variable node update rules $\Phi_{a\to i}$ depend on $a\to i$ only through $|\partial a|$ and $\widehat{J}_a$. With a slight abuse of notation, we shall denote the update functions as:

$$\Psi_{i\to a}(\{\widehat{\nu}_{b\to i}:\, b\in\partial i\setminus a\})=\Psi_l(\widehat{\nu}_1,\dots,\widehat{\nu}_l;J_i)\,, \tag{14.69}$$

$$\Phi_{a\to i}(\{\nu_{j\to a}:\, j\in\partial a\setminus i\})=\Phi_k(\nu_1,\dots,\nu_k;\widehat{J}_a)\,, \tag{14.70}$$

where we let $l\equiv|\partial i|-1$, $k\equiv|\partial a|-1$, $\{\widehat{\nu}_1,\dots,\widehat{\nu}_l\}\equiv\{\widehat{\nu}_{b\to i}:\, b\in\partial i\setminus a\}$ and $\{\nu_1,\dots,\nu_k\}\equiv\{\nu_{j\to a}:\, j\in\partial a\setminus i\}$. A similar notation will be used for the decision rule $\widehat{\Psi}$.

Exercise 14.15 Let $G=(V,E)$ be a uniformly random graph with $M=N\alpha$ edges over N vertices, and let λ_i, $i\in V$ be i.i.d. random variables uniform in $[0,\bar\lambda]$. Recall that an independent set for G is a subset of the vertices $S\subseteq V$ such that if $i,j\in S$, then (ij) is not an edge. Consider the following weighted measure over independent sets

$$\mu(S)=\frac{1}{Z}\,\mathbb{I}(S\text{ is an independent set})\prod_{i\in S}\lambda_i\,. \tag{14.71}$$

(*a*) Write the distribution $\mu(S)$ as a graphical model with binary variables and define the corresponding factor graph.

(*b*) Describe the BP algorithm to compute its marginals.

(*c*) Show that this model is a random graphical model in the sense defined above.

14.6.2 Density evolution equations

Consider a random graphical model, with factor graph $G=(V,F,E)$ and let (i,a) be a uniformly random edge in G. Let $\nu^{(t)}_{i\to a}$ be the message sent by the BP algorithm in iteration t along edge (i,a). We assume that the initial messages $\nu^{(0)}_{i\to a}$, $\widehat{\nu}^{(0)}_{a\to i}$ are i.i.d. random variables, with distribution independent of N. A considerable amount of information is contained in the distribution of $\nu^{(t)}_{i\to a}$ and $\widehat{\nu}^{(t)}_{a\to i}$ with respect to the model realization. We are interested in characterizing these distributions in the large

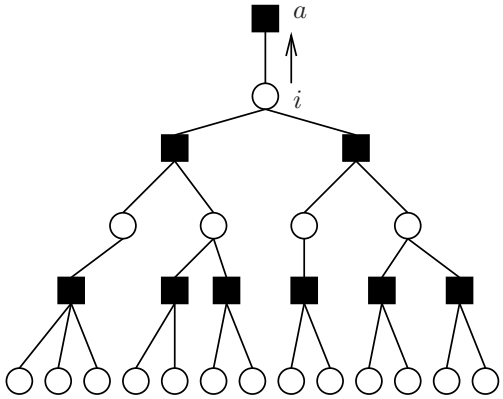

Fig. 14.8 A radius 2 directed neighborhood $\mathsf{B}_{i\to a,2}(F)$.

system limit $N \to \infty$. Our analysis will assume that both the message alphabet M and the node labels alphabet J are subsets of $\mathbb{R}^d$ for some fixed d, and that the update functions $\Psi_{i\to a}$, $\Phi_{a\to i}$ are continuous with respect to the usual topology of $\mathbb{R}^d$.

It is convenient to introduce the **directed neighborhood** of radius t of the directed edge $i \to a$: $\mathsf{B}_{i\to a,t}(G)$. This is defined as the subgraph of G that includes all the variable nodes which can be reached from i through a non-reversing path of length at most t, whose first step *is not* the edge (i, a). It includes as well all the function nodes connected only to the above specified variable nodes- see Fig. 14.8. Let us consider, to be definite, the case where G is a random factor graph from the $\mathbb{D}_N(\Lambda, P)$ ensemble. Then $\mathsf{B}_{i\to a,t}(F)$ converges in distribution, when $N \to \infty$, to the random tree ensemble $\mathbb{T}_t(\Lambda, P)$ defined in Sec. 9.5.1.

For illustrative reasons, we shall occasionally add a 'root edge' as $i \to a$ in Fig. 14.8.

Exercise 14.16 Consider a random graph from the regular $\mathbb{D}_N(\Lambda, P)$ ensemble with $\Lambda_2 = 1$, $P_3 = 1$ (each variable node has degree 2 and each function node degree 3). The three possible radius-1 directed neighborhoods appearing in such factor graphs are depicted in Fig. 14.9.

(*a*) Show that the probability that a given edge (i, a) has neighborhoods as in (B) or (C) is $O(1/N)$.

(*b*) Deduce that $\mathsf{B}_{i\to a,1}(F) \stackrel{\mathrm{d}}{\to} \mathsf{T}_1$ where T_1 is distributed according to the tree model $\mathbb{T}_1(2, 3)$ (i.e. it is the tree on Fig. 14.9, (A)).

(*c*) Discuss the case of a radius-t neighborhood.

For our purposes it is necessary to include in the description of the neighborhood $\mathsf{B}_{i\to a,t}(F)$, the value of the labels $J_i, \widehat{J}_b$ for function nodes b in this neighborhood. It is understood that the tree model $\mathbb{T}_t(\Lambda, P)$ includes labels as well: these have to be drawn as i.i.d. random variables independent of the tree and with the same distribution as in the original graphical model.

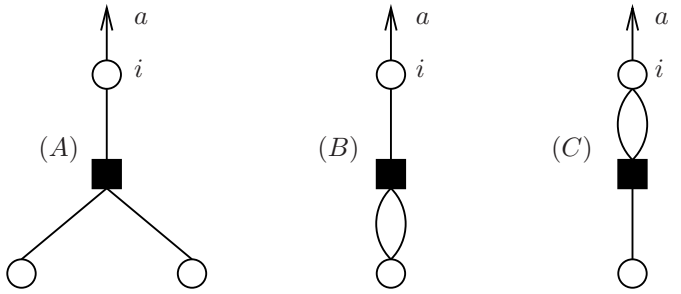

Fig. 14.9 The three possible radius–1 directed neighborhoods in a random factor graph from the regular $\mathbb{D}_N(2,3)$ graph ensemble.

Now consider the message $\nu^{(t)}_{i\to a}$. This is a function of the factor graph G, of the labels $\{J_j\}$, $\{\widehat{J}_b\}$ and of the initial condition $\{\nu^{(0)}_{j\to b}\}$. However, a moment of thought shows that its dependence on G and on the labels occurs only through the radius-$(t+1)$ directed neighborhood $\mathsf{B}_{i\to a,t+1}(F)$. Its dependence on the initial condition is only through the messages $\nu^{(0)}_{j\to b}$ for $j,b\in\mathsf{B}_{i\to a,t}(F)$.

In view of the above discussion, let us pretend for a moment that the neighborhood of (i,a) *is* a random tree T_{t+1} with distribution $\mathbb{T}_{t+1}(\Lambda,P)$. We define $\nu^{(t)}$ to be the message passed through the root edge of such a random neighborhood after t message passing iterations. Since $\mathsf{B}_{i\to a,t+1}(F)$ converges in distribution to the tree T_{t+1}, we find that[5] $\nu^{(t)}_{i\to a}\stackrel{\mathrm{d}}{\to}\nu^{(t)}$ as $N\to\infty$.

We have shown that, as $N\to\infty$, the distribution of $\nu^{(t)}_{i\to a}$ converges to the one of a well defined (N-independent) random variable $\nu^{(t)}$. The next step consists in finding a recursive characterization of $\nu^{(t)}$. Consider a random tree from the $\mathbb{T}_r(\Lambda,P)$ ensemble and let $j\to b$ be an edge directed towards the root, at distance d from it. The directed subtree rooted at $j\to b$ is distributed according to $\mathbb{T}_{r-d}(\Lambda,P)$. Therefore the message passed through it after $r-d-1$ (or more) iterations is distributed as $\nu^{(r-d-1)}$. The degree of the root variable node i (including the root edge) has distribution λ_l. Each check node connected to i has a number of other neighbors (distinct from i) which is a random variable distributed according to ρ_k. These facts imply the following distributional equations for $\nu^{(t)}$ and $\widehat{\nu}^{(t)}$:

$$\nu^{(t+1)}\stackrel{\mathrm{d}}{=}\Psi_l(\widehat{\nu}^{(t)}_1,\dots,\widehat{\nu}^{(t)}_l;J)\,,\qquad \widehat{\nu}^{(t)}\stackrel{\mathrm{d}}{=}\Phi_k(\nu^{(t)}_1,\dots,\nu^{(t)}_k;\widehat{J})\,. \tag{14.72}$$

Here $\widehat{\nu}^{(t)}_b$, $b\in\{1,\dots,l-1\}$ are independent copies of $\widehat{\nu}^{(t)}$, and $\nu^{(t)}_j$, $j\in\{1,\dots,k-1\}$ are independent copies of $\nu^{(t)}$. As for l and k, they are independent random integers distributed, respectively, according to λ_l and ρ_k, $\widehat{J}$ is distributed as $P^{(k)}_{\widehat{J}}$ and J is

[5]The mathematically suspicious reader may wonder about the topology we are assuming on the message space space. In fact no assumption is necessary if the distribution of labels $J_i,\widehat{J}_a$ is independent of N. If it is N dependent but converges, then the topology must be such that the messages updates are continuous with respect to it.

distributed as P_J. It is understood that the recursion is initiated with $\nu^{(0)} \stackrel{\mathrm{d}}{=} \nu^{(0)}_{i\to a}$, $\widehat{\nu}^{(0)} \stackrel{\mathrm{d}}{=} \widehat{\nu}^{(0)}_{a\to i}$.

In coding theory, the equations (14.72), or sometimes the sequence of random variables $\{\nu^{(t)}, \widehat{\nu}^{(t)}\}$, are referred to as **density evolution**. In probabilistic combinatorics, they are also called **recursive distributional equations**. We have proved the following characterization of the messages distribution:

Proposition 14.9 *Consider a random graphical model satisfying assumptions 1-4 in Section 14.6.1. Let $t \geq 0$ and (ia) be a uniformly random edge in the factor graph. Then, as $N \to \infty$, the message $\nu^{(t)}_{i\to a}$ (respectively $\widehat{\nu}^{(t)}_{i\to a}$) converges in distribution to the random variable $\nu^{(t)}$ (respectively $\widehat{\nu}^{(t)}$) defined through the density evolution equations (14.72).*

We shall discuss several applications of density evolution in the following chapters. Here we just mention that it allows to compute the asymptotic distribution of message passing decisions at a uniformly random site i. Recall that the general message passing decision after t iterations is taken using the rule (14.67), with $\widehat{\Psi}_i(\{\widehat{\nu}_b\}) = \widehat{\Psi}_l(\widehat{\nu}_1, \dots, \widehat{\nu}_l; J_i)$ (where $l \equiv |\partial i|$). Arguing as in the previous paragraphs it is easy to show that, in the large N limit, $\nu^{(t)}_i \stackrel{\mathrm{d}}{\to} \nu^{(t)}$, where the random variable $\nu^{(t)}$ is distributed according to:

$$\nu^{(t)} \stackrel{\mathrm{d}}{=} \widehat{\Psi}_l(\widehat{\nu}^{(t-1)}_1, \dots, \widehat{\nu}^{(t-1)}_l; J)\,. \tag{14.73}$$

As above $\widehat{\nu}^{(t-1)}_1, \dots, \widehat{\nu}^{(t-1)}_l$ are i.i.d. copies of $\widehat{\nu}^{(t-1)}$, J is an independent copy of the variable node label J_i, and l is a random integer distributed according to Λ_l.

14.6.3 The replica symmetric cavity method

The replica symmetric (RS) cavity method of statistical mechanics adopts a point of view which is very close to the previous one, but less algorithmic. Instead of considering the BP update rules as an iterative message passing rule, it focuses on the fixed point BP equations themselves.

The idea is to compute the partition function recursively, by adding one variable node at a time. Equivalently one may think of taking one variable node out of the system and computing the change in the partition function. The name of the method comes exactly from this image: one digs a 'cavity' in the system.

As an example, take the original factor graph, delete the factor node a and all the edges incident on it. If the graph is a tree, this procedure separates it into $|\partial a|$ disconnected trees. Consider now the tree-graphical model described by the connected component containing the variable $j \in \partial a$. Denote the corresponding partition function, when the variable j is fixed to the value x_j, by $Z_{j\to a}(x_j)$. These partial partition functions can be computed iteratively as:

$$Z_{j\to a}(x_j) = \prod_{b\in\partial j\setminus a} \left[\sum_{\underline{x}_{\partial b\setminus j}} \psi_b(\underline{x}_{\partial b}) \prod_{k\in\partial b\setminus j} Z_{k\to b}(x_k) \right]\,. \tag{14.74}$$

The equations obtained by letting $j \to b$ be a generic directed edge in G, are called **cavity equations**, or **Bethe equations**.

The cavity equations are mathematically identical to the BP equations, with two important conceptual differences: (*i*) One is naturally led to think that the equations (14.74) must have a fixed point, and to give special importance to it; (*ii*) The partial partition functions are unnormalized messages, and, as we will see in Chapter **??**, their normalization provides a useful information. The relation between BP messages and partial partition functions is

$$\nu_{j\to a}(x_j) = \frac{Z_{j\to a}(x_j)}{\sum_y Z_{j\to a}(y)} \,. \tag{14.75}$$

Within the cavity approach, the **replica symmetry assumption** consists in pretending that, for random graphical models as introduced above, and in the large N limit:

1. There exists a solution (or quasi-solution[6]) to these equations.
2. This solution provides good approximations of the marginals of the graphical model.
3. The messages in this solution are distributed according to a density evolution fixed point.

The last statement amounts to assuming that the normalized variable-to-factor messages $\nu_{i\to a}$, cf. Eq. (14.75), converge in distribution to a random variable ν, that solves the distributional equations:

$$\nu \stackrel{\rm d}{=} \Psi(\widehat{\nu}_1, \dots, \widehat{\nu}_{k-1}; J) \,, \qquad \widehat{\nu} \stackrel{\rm d}{=} \Phi(\nu_1, \dots, \nu_{l-1}; \widehat{J}) \,. \tag{14.76}$$

Here we use the same notations as in Eq. (14.72): $\widehat{\nu}_b$, $b \in \{1, \dots, l-1\}$ are independent copies of $\widehat{\nu}^{(t)}$; $\nu_j^{(t)}$, $j \in \{1, \dots, k-1\}$ are independent copies of $\nu^{(t)}$; l and k are independent random integers distributed, respectively, according to λ_l and ρ_k; J, $\widehat{J}$ are distributed as the variable and function nodes labels J_i, $\widehat{J}_a$.

Using the distributions of ν and $\widehat{\nu}$, the expected Bethe free-entropy per variable $\mathbb{F}/N$ can be computed by taking the expectation of Eq. (14.27). The result is:

$$\mathrm{f}^{\rm RS} = \mathrm{f}_{\rm v}^{\rm RS} + n_{\rm f}\mathrm{f}_{\rm f}^{\rm RS} - n_{\rm e}\mathrm{f}_{\rm e}^{\rm RS} \tag{14.77}$$

where $n_{\rm f}$ is the average number of function nodes per variable, and $n_{\rm e}$ is the average number of edges per variable: In the $\mathbb{D}_N(\Lambda, P)$ ensemble one has $n_{\rm f} = \Lambda'(1)/P'(1)$ and $n_{\rm e} = \Lambda'(1)$; Within the $\mathbb{G}_N(K, \alpha)$ ensemble, $n_{\rm f} = \alpha$ and $n_{\rm e} = K\alpha$. The contributions of variable nodes $\mathrm{f}_{\rm v}^{\rm RS}$, function nodes $\mathrm{f}_{\rm f}^{\rm RS}$, and edges $\mathrm{f}_{\rm e}^{\rm RS}$ are:

[6] A quasi-solution is a set of messages $\nu_{j\to a}$ such that the average difference between the left and right hand sides of the BP equations goes to zero in the large N limit

$$
\begin{aligned}
\mathrm{f}_{\mathrm{v}}^{\mathrm{RS}} &= \mathbb{E}_{l,J,\{\widehat{\nu}\}} \log \left[\sum_x \psi(x;J)\, \widehat{\nu}_1(x) \cdots \widehat{\nu}_l(x) \right] \,, \\
\mathrm{f}_{\mathrm{f}}^{\mathrm{RS}} &= \mathbb{E}_{k,\widehat{J},\{\nu\}} \log \left[\sum_{x_1,\ldots,x_k} \psi^{(k)}(x_1,\ldots,x_k;\widehat{J})\, \nu_1(x_1) \cdots \nu_k(x_k) \right] \,, \\
\mathrm{f}_{\mathrm{e}}^{\mathrm{RS}} &= \mathbb{E}_{\nu,\widehat{\nu}} \log \left[\sum_x \nu(x)\widehat{\nu}(x) \right] \,. \qquad (14.78)
\end{aligned}
$$

In these expressions, $\mathbb{E}$ denotes expectation with respect to the random variables in subscript. For instance, if G is distributed according to the $\mathbb{D}_N(\Lambda, P)$ ensemble, $\mathbb{E}_{l,J,\{\widehat{\nu}\}}$ implies that l is drawn from distribution Λ, J is drawn from P_J, and $\widehat{\nu}_1, \ldots \widehat{\nu}_l$ are l independent copies of the random variable $\widehat{\nu}$.

Instead of estimating the partition function, the cavity method can be used to compute the ground state energy. One then uses min-sum like messages instead of those in (14.74). The method is then called the 'energetic cavity method', we leave to the reader the task of writing the corresponding average ground state energy per variable.

14.6.4 Numerical methods

Generically, the RS cavity equations (14.76), as well as density evolution (14.72), cannot be solved in close form, and one uses numerical methods to estimate the distribution of the random variables ν, $\widehat{\nu}$. Here we limit ourselves to describing a stochastic approach that has the advantage of being extremely versatile and simple to implement. It has been used in coding theory under the name of 'sampled density evolution' or 'Monte Carlo', and is known in statistical physics as **population dynamics**, a name which we shall adopt in the following.

The idea is to approximate the distribution of ν (or $\widehat{\nu}$) through a sample of (ideally) N i.i.d. copies of ν (respectively $\widehat{\nu}$). As N gets large, the empirical distribution of such a sample should converge to the actual distribution of ν (or $\widehat{\nu}$). We shall call the sample $\{\nu_i\} \equiv \{\nu_1, \ldots, \nu_N\}$ (or $\{\widehat{\nu}_i\} \equiv \{\widehat{\nu}_1, \ldots, \widehat{\nu}_N\}$) a **population**.

The algorithm is described by the pseudo-code below. As inputs, it requires the population size N, the maximum number of iterations T and a specification of the ensemble of (random) graphical models. The latter consists in a description of the (edge perspective) degree distributions λ, ρ , of the variable node labels P_J, and of the factor node labels $P_{\widehat{J}}^{(k)}$

POPULATION DYNAMICS (Model ensemble, Size N, Iterations T)

1: Initialize $\{\nu_i^{(0)}\}$;
2: **for** $t = 1, \dots, T$:
3: **for** $i = 1, \dots, N$:
4: Draw an integer k with distribution ρ;
5: Draw $i(1), \dots, i(k-1)$ uniformly in $\{1, \dots, N\}$;
6: Draw $\widehat{J}$ with distribution $P_{\widehat{J}}^{(k)}$;
7: Set $\widehat{\nu}_i^{(t)} = \Phi_k(\nu_{i(1)}^{(t-1)}, \dots, \nu_{i(k-1)}^{(t-1)}; \widehat{J})$;
8: **end;**
9: **for** $i = 1, \dots, N$:
10: Draw an integer l with distribution λ;
11: Draw $i(1), \dots, i(l-1)$ uniformly in $\{1, \dots, N\}$;
12: Draw J with distribution P_J;
13: Set $\nu_i^{(t)} = \Psi_l(\widehat{\nu}_{i(1)}^{(t)}, \dots, \widehat{\nu}_{i(l-1)}^{(t)}; J)$;
14: **end;**
15: **end;**
16: **return** $\{\nu_i^{(T)}\}$ and $\{\widehat{\nu}_i^{(T)}\}$.

In step 1 the initialization is done by drawing $\nu_1^{(0)}, \dots, \nu_N^{(0)}$ independently with the same distribution P that was used for the initialization of BP.

It is not hard to show that, for any fixed T, the empirical distribution of $\{\nu_i^{(T)}\}$ (respectively $\{\widehat{\nu}_i^{(T)}\}$) converges, as $N \to \infty$ to the distribution of the density evolution random variable $\nu^{(t)}$ ($\widehat{\nu}^{(t)}$). The limit $T \to \infty$ is trickier. Let us first assume that density evolution has a unique fixed point, and $\nu^{(t)}$, $\widehat{\nu}^{(t)}$ converges to this fixed point. Then we expect the empirical distribution of $\{\nu_i^{(T)}\}$ to converge to this fixed point, also if the $N \to \infty$ limit is taken after $T \to \infty$. Finally, when density evolution has more than a fixed point, which is probably some of the most interesting case, the situation is even more subtle. The population $\{\nu_i^{(T)}\}$ evolves according to a large, but finite dimensional Markov chain. Therefore (under some technical conditions) the distribution of the population is expected to converge to the unique fixed point of this Markov chain. This seems to imply that population dynamics cannot describe the multiple fixed points of density evolution. Luckily, the convergence of population dynamics to its unique fixed point appears to happen on a time scale that increases very rapidly with N. For large N and on moderate time scales T, it converges instead to one of several 'quasi-fixed points' that correspond to the density evolution fixed points.

In practice, one can monitor the effective convergence of the algorithm by computing, after any number of iterations t, averages of the form

$$\langle \varphi \rangle_t \equiv \frac{1}{N} \sum_{i=1}^{N} \varphi(\nu_i^{(t)}) \,, \tag{14.79}$$

for a smooth function $\varphi : \mathfrak{M}(\mathcal{X}) \to \mathbb{R}$. If these averages are well settled (up to statistical

fluctuations of order $1/\sqrt{N}$), this is interpreted as a signal that the iteration has converged to a 'quasi-fixed point.'

The populations produced by the above algorithm can be used to to estimate expectation with respect to the density evolution random variables ν, $\widehat{\nu}$. For instance, the expression in Eq. (14.79) is an estimate for $\mathbb{E}\{\varphi(\nu)\}$. When $\varphi = \varphi(\nu_!, \dots, \nu_l)$ is a function of l i.i.d. copies of ν, the above formula is modified as

$$\langle \varphi \rangle_t \equiv \frac{1}{R} \sum_{n=1}^{R} \varphi(\nu^{(t)}_{i_n(1)}, \dots, \nu^{(t)}_{i_n(l)}) \, . \tag{14.80}$$

Here R is a large number (typically of the same order as N), and $i_n(1), \dots, i_n(l)$ are i.i.d. indices in $\{1, \dots, N\}$. Of course such estimates will be reasonable only if $l \ll N$.

A particularly important example is the computation of the free entropy (14.77). Each of the terms $\mathrm{f}_{\mathrm{v}}^{\mathrm{RS}}$, $\mathrm{f}_{\mathrm{f}}^{\mathrm{RS}}$ and $\mathrm{f}_{\mathrm{e}}^{\mathrm{RS}}$ can be estimated as in Eq. (14.80). The precision of these estimates can be improved by repeating the computation for several iterations and averaging the result.

Notes

Belief propagation equations have been rediscovered several times. They were developed by Pearl (Pearl, 1988) as exact algorithm for probabilistic inference in acyclic Bayesian networks. In the early 60's, Gallager had introduced them as an iterative procedure for decoding low density parity check codes (Gallager, 1963). Gallager described several message passing procedures and, among them, the sum-product algorithm. Within coding theory, the basic idea of this algorithm was rediscovered in several works in the 90's, and, in particular, in (Berrou and Glavieux, 1996).

In the physics context, the history is even longer. In 1935, Bethe used a free-energy functional written in terms of pseudo-marginals to approximate the partition function of the ferromagnetic Ising model (Bethe, 1935). Bethe equations were of the simple form discussed Exercise 14.10, because of the homogeneity (translation invariance) of the underlying model. Their generalization to inhomogeneous systems, which has a natural algorithmic interpretation, waited until the application of Bethe's method to spin glasses (Thouless, Anderson and Palmer, 1977; Klein, Schowalter and Shukla, 1979; Katsura, Inawashiro and Fujiki, 1979; Morita, 1979; Nakanishi, 1981).

The review paper (Kschischang, Frey and Loeliger, 2001) gives a general overview of belief propagation in the factor graphs framework. The role of the distributive property, mentioned in Exercise 14.8, is emphasized in (Aji and McEliece, 2000). On tree graphs, belief propagation can be regarded as an instance of the junction-tree algorithm (Lauritzen, 1996). This algorithm constructs a tree from the graphical model under study, by grouping some of its variables. Belief propagation is then applied to this tree.

Although implicit in these earlier works, the equivalence between BP, Bethe approximation, and sum-product algorithm was only recognized in the 90's. The turbo-decoding and sum-product algorithm were shown to be instances of BP in (McEliece, MacKay and Cheng, 1998). A variational derivation of the turbo decoding algorithm was proposed in (Montanari and Sourlas, 2000). The equivalence between BP and

Bethe approximation was first put forward in (Kabashima and Saad, 1998) and, in a more general setting, in (Yedidia, Freeman and Weiss, 2001; Yedidia, Freeman and Weiss, 2005).

The last paper proved, in particular, the variational formulation in Proposition 14.8. This suggests to look for fixed points of BP by seeking directly stationary points of the Bethe free-entropy, without iterating the BP equations. An efficient such procedure, based on the observation that the Bethe free-entropy can be written as the difference between a convex and a concave function, was proposed in (Yuille, 2002). An alternative approach consists in constructing convex surrogates of the Bethe free-energy (Wainwright, Jaakkola and Willsky, 2005*b*; Wainwright, Jaakkola and Willsky, 2005*a*), which allow to define provably convergent message passing procedures.

Bethe approximation can also be regarded as a first step in a hierarchy of variational methods describing exactly larger and larger clusters of variables. This point of view was first developed in (Kikuchi, 1951), leading to the so called 'cluster variational method' in physics. The algorithmic version of this approach is referred to as 'generalized BP,' and is described in details in (Yedidia, Freeman and Weiss, 2005).

The analysis of iterative message passing algorithms on random graphical models dates back to (Gallager, 1963). These ideas were developed into a systematic method, also thanks to efficient numerical techniques, in (Richardson and Urbanke, 2001*b*) who coined the name 'density evolution.' The point of view taken in this book is however closer to the one of 'local weak convergence' (Aldous and Steele, 2003).

In physics, the replica symmetric cavity method for sparse random graphical models, was first discussed in (Mézard and Parisi, 1987). The use of population dynamics first appeared in (Abou-Chacra, Anderson and Thouless, 1973), and was further developed for spin glasses in (Mézard and Parisi, 2001), but this paper mainly deals with RSB effects which will be the object of Ch. **??**.

15
Decoding with belief propagation

As we have already seen, symbol MAP decoding of error correcting codes can be regarded as a statistical inference problem. It is a very natural idea to accomplish this task using belief propagation (BP). For properly constructed codes (in particular LDPC ensembles), this approach has low complexity while achieving very good performances.

However, it is clear that an error correcting code cannot achieve good performances unless the associated factor graph has loops. As a consequence, belief propagation has to be regarded only as an approximate inference algorithm in this context. A major concern of the theory is to establish conditions for its optimality, and, more generally, the relation between message passing and optimal (exact symbol MAP) decoding.

In this chapter we discuss belief propagation decoding of the LDPC ensembles introduced in Chapter 11. The message passing approach can be generalized to several other applications within information and communication theory: other code ensembles, source coding, channels with memory, etc.... Here we shall keep to the 'canonical' example of channel coding as most of the theory has been developed in this context.

BP decoding is defined in Section 15.1. One of the main tools in the analysis is the 'density evolution' method that we discuss in Section 15.2. This allows to determine the threshold for reliable communication under BP decoding, and to optimize accordingly the code ensemble. The whole process is considerably simpler for the erasure channel, which is treated in Section 15.3. Finally, Section 15.4 explains the relation between optimal (MAP) decoding and BP decoding in the large block-length limit: the two approaches can studied within the unified framework base on the Bethe free-energy.

15.1 BP decoding: the algorithm

In this chapter, we shall consider communication over a **binary input, output symmetric, memoryless channel (BMS)**. This is a channel in which the transmitted codeword is binary, $\underline{x} \in \{0,1\}^N$, and the output $\underline{y}$ is a sequence of N letters y_i from an alphabet $\mathcal{Y} \subset \mathbb{R}$. The probability of receiving letter y when bit x is sent, $Q(y|x)$, enjoys the symmetry property $Q(y|0) = Q(-y|1)$.

Let us suppose that a LDPC error correcting code is used in this communication. The conditional probability for the channel input being $\underline{x} \in \{0,1\}^N$ given the output $\underline{y}$ is $\mathbb{P}(\underline{x}|\underline{y}) = \mu_{\underline{y}}(\underline{x})$, where

$$\mu_{\underline{y}}(\underline{x}) = \frac{1}{Z(\underline{y})} \prod_{i=1}^{N} Q(y_i|x_i) \prod_{a=1}^{M} \mathbb{I}(x_{i_1^a} \oplus \cdots \oplus x_{i_{k(a)}^a} = 0)\,, \tag{15.1}$$

The factor graph associated with this distribution is the usual one: an edge joins a variable node i to a check node a whenever the variable x_i appears in the a-th parity check equation.

Messages $\nu_{i\to a}(x_i)$, $\widehat{\nu}_{a\to i}(x_i)$, are exchanged along the edges. We shall assume a parallel updating of BP messages, as introduced in Sec. 14.2:

$$\nu_{i\to a}^{(t+1)}(x_i) \cong Q(y_i|x_i) \prod_{b\in\partial i\setminus a} \widehat{\nu}_{b\to i}^{(t)}(x_i)\,, \tag{15.2}$$

$$\widehat{\nu}_{a\to i}^{(t)}(x_i) \cong \sum_{\{x_j\}} \mathbb{I}(x_i \oplus x_{j_1} \oplus \cdots \oplus x_{j_{k-1}} = 0) \prod_{j\in\partial a\setminus i} \nu_{j\to a}^{(t)}(x_j)\,, \tag{15.3}$$

where we used the notation $\partial a \equiv \{i, j_1, \dots, j_{k-1}\}$, and the symbol $\cong$ denotes as before 'equality up to a normalization constant'. We expect that the asymptotic performances at large t and large N of such BP decoding, for instance its asymptotic bit error rate, should be insensitive to the precise update schedule. On the other hand, this schedule can have an important influence on the speed of convergence, and on performances at moderate N. Here we shall not address these issues.

The BP estimate for the marginal distribution at node i at time t, also called 'belief' or **'soft decision'**, is

$$\nu_i^{(t)}(x_i) \;\cong\; Q(y_i|x_i) \prod_{b\in\partial i} \widehat{\nu}_{b\to i}^{(t-1)}(x_i)\,. \tag{15.4}$$

Based on this estimate, the optimal BP decision for bit i at time t, sometimes called **'hard decision'**, is

$$\widehat{x}_i^{(t)} = \arg\max_{x_i} \nu_i^{(t)}(x_i)\,. \tag{15.5}$$

In order to fully specify the algorithm, one should address two more issues: (1) How are the messages initialized, and (2) After how many iterations t, the hard decision (15.5) is taken.

In practice, one usually initializes the messages to $\nu_{i\to a}^{(0)}(\mathtt{0}) = \nu_{i\to a}^{(0)}(\mathtt{1}) = 1/2$. One alternative choice, that is sometimes useful for theoretical reasons, is to take the messages $\nu_{i\to a}^{(0)}(\,\cdot\,)$ as independent random variables, for instance by choosing $\nu_{i\to a}^{(0)}(\mathtt{0})$ uniformly on $[0,1]$.

As for the number of iterations, one would like to have a stopping criterion. In practice, a convenient criterion is to check whether $\underline{\widehat{x}}^{(t)}$ is a codeword, and to stop if this is the case. If this condition is not fulfilled, the algorithm is stopped after a fixed number of iterations $t_{\max}$. On the other hand, for the purpose of performance analysis, we shall rather fix $t_{\max}$ and assume that belief propagation is run always for $t_{\max}$ iterations, regardless whether a valid codeword is reached at an earlier stage.

Since the messages are distributions over binary valued variables, we parameterize them by the log-likelihoods:

$$h_{i\to a} = \frac{1}{2}\,\log\frac{\nu_{i\to a}(\mathtt{0})}{\nu_{i\to a}(\mathtt{1})}\,, \qquad u_{a\to i} = \frac{1}{2}\,\log\frac{\widehat{\nu}_{a\to i}(\mathtt{0})}{\widehat{\nu}_{a\to i}(\mathtt{1})}\,. \tag{15.6}$$

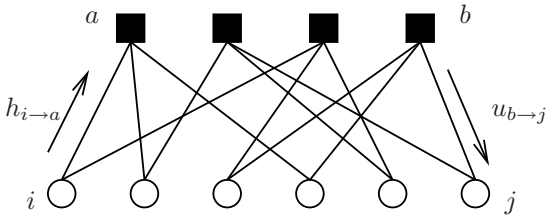

Fig. 15.1 Factor graph of a (2,3) regular LDPC code, and notation for the belief propagation messages.

We further introduce the a-priori log-likelihood for bit i, given the received message y_i:

$$B_i = \frac{1}{2}\,\log\frac{Q(y_i|0)}{Q(y_i|1)}\,. \tag{15.7}$$

For instance if communication takes place over a BSC channel with flip probability p, one has $B_i = \frac{1}{2}\log\frac{1-p}{p}$ on variable nodes which have received $y_i = 0$, and $B_i = -\frac{1}{2}\log\frac{1-p}{p}$ on those with $y_i = 1$. The BP update equations (15.2), (15.3) read in this notation (see Fig. 15.1):

$$h_{i\to a}^{(t+1)} = B_i + \sum_{b\in\partial i\setminus a} u_{b\to i}^{(t)}\,, \qquad u_{a\to i}^{(t)} = \operatorname{atanh}\Big\{\prod_{j\in\partial a\setminus i}\tanh h_{j\to a}^{(t)}\Big\}\,. \tag{15.8}$$

The hard-decision decoding rule depends on the overall BP log-likelihood

$$h_i^{(t+1)} = B_i + \sum_{b\in\partial i} u_{b\to i}^{(t)}\,, \tag{15.9}$$

and is given by (using for definiteness a fair coin outcome in case of a tie):

$$\widehat{x}_i^{(t)}(\underline{y}) = \begin{cases} 0 & \text{if } h_i^{(t)} > 0,\\ 1 & \text{if } h_i^{(t)} < 0,\\ 0 \text{ or } 1 & \text{with probability } 1/2 \text{ if } h_i^{(t)} = 0. \end{cases} \tag{15.10}$$

15.2 Analysis: density evolution

Let us study BP decoding of random codes from the $\mathrm{LDPC}_N(\Lambda, P)$ ensemble in the large block-length limit. The code ensemble is specified by the degree distributions of variable nodes $\Lambda = \{\Lambda_l\}$ and of check nodes, $P = \{P_k\}$. We assume for simplicity that messages are initialized to $u_{a\to i}^{(0)} = 0$.

Because of the symmetry of the channel, under the above hypotheses, the bit (or block) error probability is independent of the transmitted codeword. The explicit derivation of this fact is outlined in Exercise 15.1 below. Thanks to this freedom, we can assume that the all-zero codeword has been transmitted. We shall first write the density evolution recursion as a special case of the one written in Sec. 14.6.2. It turns

out that this recursion can be analyzed in quite some detail, and in particular one can show that the decoding performance improves as t increases. The analysis hinges on two important properties of BP decoding and density evolution, related to the notions of 'symmetry' and 'physical degradation'.

Exercise 15.1 Independence of the transmitted codeword. Assume the codeword $\underline{x}$ has been transmitted and let $B_i(\underline{x})$, $u^{(t)}_{a\to i}(\underline{x})$, $h^{(t)}_{i\to a}(\underline{x})$ be the corresponding channel log-likelihoods and messages. Because of the randomness in the channel realization, they are random variables. Let furthermore $\sigma_i = \sigma_i(\underline{x}) = +1$ if $x_i = 0$, and $= -1$ otherwise.

(*a*) Prove that the distribution of $\sigma_i B_i$ is independent of $\underline{x}$.

(*b*) Use the equations (15.8) to prove by induction over t that the (joint) distribution of $\{\sigma_i h^{(t)}_{i\to a}, \sigma_i u^{(t)}_{a\to i}\}$ is independent of $\underline{x}$.

(*c*) Use Eq. (15.9) to show that the distribution of $\{\sigma_i h^{(t)}_i\}$ is independent of $\underline{x}$ for any $t \geq 0$. Finally, prove that the distribution of the 'error vector' $\underline{z}^{(t)} \equiv \underline{x} \oplus \widehat{\underline{x}}^{(t)}(\underline{y})$ is independent of $\underline{x}$ as well. Write the bit and block error rate in terms of the distribution of $\underline{z}^{(t)}$.

15.2.1 Density evolution equations

Let us consider the distribution of messages after a fixed number t of iterations. As we saw in Sec. 14.6.2, in the large N limit, the directed neighborhood of any given edge is with high probability a tree, whose distribution converges to the model $\mathbb{T}_t(\Lambda, P)$. This implies the following recursive distributional characterization for $h^{(t)}$ and $u^{(t)}$:

$$h^{(t+1)} \stackrel{\mathrm{d}}{=} B + \sum_{b=1}^{l-1} u^{(t)}_b\,, \qquad u^{(t)} \stackrel{\mathrm{d}}{=} \operatorname{atanh}\Big\{\prod_{j=1}^{k-1} \tanh h^{(t)}_j\Big\}\,. \tag{15.11}$$

Here $u^{(t)}_b$, $b \in \{1,\dots,l-1\}$ are independent copies of $u^{(t)}$, $h^{(t)}_j$, $j \in \{1,\dots,k-1\}$ are independent copies of $h^{(t)}$, l and k are independent random integers distributed, respectively, according to λ_l and ρ_k. Finally, $B = \frac{1}{2}\log\frac{Q(y|0)}{Q(y|1)}$ where y is independently distributed according to $Q(y|0)$. The recursion is initialized with $u^{(0)} = 0$.

Let us finally consider the BP log-likelihood at site i. The same arguments as above imply $h^{(t)}_i \stackrel{\mathrm{d}}{\to} h^{(t)}_*$, where the distribution of $h^{(t)}_*$ is defined by

$$h^{(t+1)}_* \stackrel{\mathrm{d}}{=} B + \sum_{b=1}^{l} u^{(t)}_b\,, \tag{15.12}$$

with l a random integer distributed according to Λ_l. In particular, if we let $\mathrm{P}^{(N,t)}_{\mathrm{b}}$ be the expected (over a $\mathrm{LDPC}_N(\Lambda, P)$ ensemble) bit error rate for the decoding rule (15.10), then:

$$\lim_{N\to\infty} \mathrm{P}^{(N,t)}_{\mathrm{b}} = \mathbb{P}\big\{h^{(t)}_* < 0\big\} + \frac{1}{2}\mathbb{P}\big\{h^{(t)}_* = 0\big\}\,. \tag{15.13}$$

The suspicious reader will notice that this statement is non-trivial, because $f(x) = \mathbb{I}(x < 0) + \frac{1}{2}\mathbb{I}(x = 0)$ is not a continuous function. We shall prove it below using the symmetry property of the distribution of $h_i^{(t)}$,which allows to write the bit error rate as the expectation of a continuous function (cf. Exercise 15.2).

15.2.2 Basic properties: 1. Symmetry

A real random variable Z (or, equivalently, its distribution) is said to be **symmetric** if

$$\mathbb{E}\{f(-Z)\} = \mathbb{E}\left\{e^{-2Z} f(Z)\right\} . \tag{15.14}$$

for any function f such that one of the expectations exists. If Z has a density $p(z)$, then the above condition is equivalent to $p(-z) = e^{-2z} p(z)$.

Symmetric variables appear naturally in the description of BMS channels:

Proposition 15.1 *Consider a BMS channel with transition probability $Q(y|x)$. Let Y be the channel output conditional to input $\mathtt{0}$ (this is a random variable with distribution $Q(y|\mathtt{0})$), and let $B \equiv \frac{1}{2}\log\frac{Q(Y|\mathtt{0})}{Q(Y|\mathtt{1})}$. Then B is a symmetric random variable.*

Conversely, if Z is a symmetric random variable, there exists a BMS channel whose log-likelihood ratio, conditioned on the input being $\mathtt{0}$, is distributed as Z.

Proof: To avoid technicalities, we prove this claim when the output alphabet $\mathcal{Y}$ is a *discrete* subset of $\mathbb{R}$. Then, using channel symmetry in the form $Q(y|\mathtt{0}) = Q(-y|\mathtt{1})$, we get

$$\begin{aligned}\mathbb{E}\{f(-B)\} &= \sum_y Q(y|\mathtt{0})\, f\left(\frac{1}{2}\log\frac{Q(y|\mathtt{1})}{Q(y|\mathtt{0})}\right) = \sum_y Q(y|\mathtt{1})\, f\left(\frac{1}{2}\log\frac{Q(y|\mathtt{0})}{Q(y|\mathtt{1})}\right) = \\ &= \sum_y Q(y|\mathtt{0})\,\frac{Q(y|\mathtt{1})}{Q(y|\mathtt{0})}\, f\left(\frac{1}{2}\log\frac{Q(y|\mathtt{0})}{Q(y|\mathtt{1})}\right) = \mathbb{E}\left\{e^{-2B} f(B)\right\} . \end{aligned} \tag{15.15}$$

We now prove the converse. Let Z be a symmetric random variable. We build a channel with output alphabet $\mathbb{R}$ as follows: Under input $\mathtt{0}$, the output is distributed as Z, and under input $\mathtt{1}$, it is distributed as $-Z$. In terms of densities

$$Q(z|\mathtt{0}) = p(z)\,, \qquad Q(z|\mathtt{1}) = p(-z)\,. \tag{15.16}$$

This is a BMS channel with the desired property. Of course this construction is not unique. □

Example 15.2 Consider the binary erasure channel BEC(ϵ). If the channel input is $\mathtt{0}$, then Y can take two values, either $\mathtt{0}$ (with probability $1-\epsilon$) or $*$ (probability ϵ). The distribution of B, $\mathbb{P}_B = (1-\epsilon)\,\delta_\infty + \epsilon\,\delta_0$, is symmetric. In particular, this is true for the two extreme cases: $\epsilon = 0$ (a noiseless channel) and $\epsilon = 1$ (a completely noisy channel: $\mathbb{P}_B = \delta_0$).

Example 15.3 Consider a binary symmetric channel BSC(p). The log-likelihood B can take two values, either $b_0 = \frac{1}{2}\log\frac{1-p}{p}$ (input 0 and output 0) or $-b_0$ (input 0 and output 1). Its distribution, $\mathbb{P}_B = (1-p)\,\delta_{b_0} + p\,\delta_{-b_0}$ is symmetric.

Example 15.4 Finally consider the binary white noise additive Gaussian channel BAWGN(σ^2). If the channel input is 0, the output Y has probability density

$$q(y) = \frac{1}{\sqrt{2\pi\sigma^2}}\,\exp\left\{-\frac{(y-1)^2}{2\sigma^2}\right\}\,, \tag{15.17}$$

i.e. it is a Gaussian of mean 1 and variance σ^2. The output density upon input 1 is determined by the channel symmetry, it is therefore a Gaussian of mean -1 and variance σ^2. The log-likelihood under output y is easily checked to be $b = y/\sigma^2$. Therefore B also has a symmetric Gaussian density, namely:

$$p(b) = \sqrt{\frac{\sigma^2}{2\pi}}\,\exp\left\{-\frac{\sigma^2}{2}\left(b-\frac{1}{\sigma^2}\right)^2\right\}\,. \tag{15.18}$$

The variables appearing in density evolution are symmetric as well. The argument is based on the symmetry of the channel log-likelihood, and the fact that symmetry is preserved by the operations in BP evolution: If Z_1 and Z_2 are two independent symmetric random variables (not necessarily identically distributed), it is straightforward to show that $Z = Z_1 + Z_2$, and $Z' = \text{atanh}[\tanh Z_1 \tanh Z_2]$ are both symmetric.

Let us consider now the communication of the all-zero codeword over a BMS channel using a LDPC code, but let us first assume that the factor graph associated with the code is a tree. We apply BP decoding with a symmetric random initial condition like e.g. $u^{(0)}_{a\to i} = 0$. The messages passed during the decoding procedure can be regarded as random variables, because of the random received symbols y_i (which yield random log-likelihoods B_i). Furthermore, messages incoming at a given node are independent since they are functions of B_i's (and of initial conditions) on disjoint subtrees. From the above remarks, and looking at the BP equations (15.8) it follows that the messages $u^{(t)}_{a\to i}$, and $h^{(t)}_{i\to a}$, as well as the overall log-likelihoods $h^{(t)}_i$ are symmetric random variables at all $t \geq 0$. Therefore:

Proposition 15.5 *Consider BP decoding of an LDPC code under the above assumptions. If* $\mathsf{B}_{i\to a,t+1}(F)$ *is a tree, then* $h^{(t)}_{i\to a}$ *is a symmetric random variable. Analogously, if* $\mathsf{B}_{i,t+1}(F)$ *is a tree, then* $H^{(t)}_i$ *is a symmetric random variable.*

Proposition 15.6 *The density evolution random variables* $\{h^{(t)}, u^{(t)}, H^{(t)}_*\}$ *are symmetric.*

Exercise 15.2 Using Proposition 15.5, and the fact that, for any finite t $\mathsf{B}_{i\to a,t+1}(F)$ is a tree with high probability as $N\to\infty$, show that

$$\lim_{N\to\infty} \mathrm{P}_{\mathrm{b}}^{(N,t)} = \lim_{N\to\infty} \mathbb{E}\left\{\frac{1}{N}\sum_{i=1}^{N} f(h_i^{(t)})\right\}, \tag{15.19}$$

where $f(x)=1/2$ for $x\le 0$ and $f(x)=e^{-2x}/2$ otherwise.

Symmetry does not hold uniquely for the BP log-likelihood, but also for the actual (MAP) log-likelihood of a bit, as shown in the exercise below.

Exercise 15.3 Consider the actual (MAP) log-likelihood for bit i (as opposed to its BP approximation). This is defined as

$$h_i(\underline{y}) = \frac{1}{2}\log\frac{\mathbb{P}\{x_i=\mathtt{0}|\underline{y}\}}{\mathbb{P}\{x_i=\mathtt{1}|\underline{y}\}}\,. \tag{15.20}$$

If we condition on the all-zero codeword being transmitted, so that $\mathbb{P}(\underline{y})=\prod_i Q(y_i|0)$, then the random variable $H_i=h_i(\underline{y})$ is symmetric. This can be shown as follows.

(*a*) Suppose that a codeword $\underline{z}\neq\underline{\mathtt{0}}$ has been transmitted, so that $\mathbb{P}(\underline{y})=\prod_i Q(y_i|z_i)$, and define in this case the random variable associated with the log-likelihood of bit x_i as: $H_i^{(\underline{z})}=h_i(\underline{y})$. Show that $H_i^{(\underline{z})}\stackrel{\mathrm{d}}{=}H_i$ if $z_i=\mathtt{0}$, and $H_i^{(\underline{z})}\stackrel{\mathrm{d}}{=}-H_i$ if $z_i=\mathtt{1}$.

(*b*) Consider the following process. A bit z_i is chosen uniformly at random. Then a codeword $\underline{z}$ is chosen uniformly at random conditioned on the value of z_i, and transmitted through a BMS channel, yielding an output $\underline{y}$. Finally, the log-likelihood $H_i^{(\underline{z})}$ is computed. Hiding the intermediate steps in a black box, this can be seen as a communication channel: $z_i\to H_i^{(\underline{z})}$. Show this is a BMS channel.

(*c*) Show that H_i is a symmetric random variable.

The symmetry property is a generalization of the Nishimori condition that we encountered in spin glasses. As can be recognized from Eq. (12.7) the Nishimori condition is satisfied if and only if for each coupling constant J, βJ is a symmetric random variable. While in spin glasses symmetry occurs only at very special values of the temperature, it holds generically for decoding. The common mathematical origin of these properties can be traced back to the structure discussed in Sec. 12.2.3.

15.2.3 Basic properties: 2. Physical degradation

It turns out that, for large blocklengths, BP decoding gets better when the number of iterations t increases (although it does not necessarily converge to the correct values). This is an extremely useful result, which does not hold when BP is applied to general

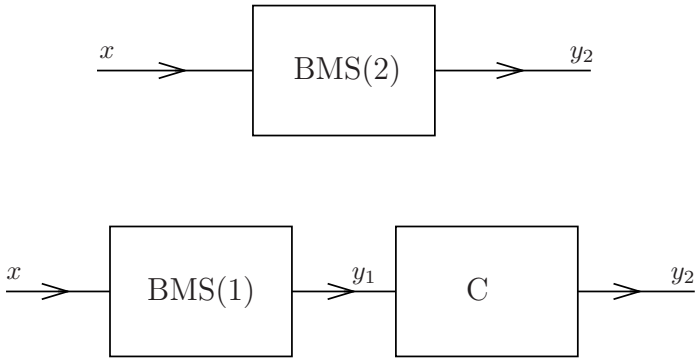

Fig. 15.2 The channel BMS(2) (top) is said to be physically degraded with respect to BMS(1) if it is equivalent to the concatenation of BMS(1) with a second channel C.

inference problems. A precise formulation of this statement is provided by the notion of physical degradation. We shall first define this notion in terms of BMS channels, and then extend it to all symmetric random variables. This allows to apply it to the random variables encountered in BP decoding and density evolution.

Let us start with the case of BMS channels. Consider two such channels, denoted as BMS(1) and BMS(2), denote by $\{Q_1(y|x)\}$, $\{Q_2(y|x)\}$, their transition matrices and by $\mathcal{Y}_1$, $\mathcal{Y}_2$, the corresponding output alphabets. We say that BMS(2) is **physically degraded** with respect to BMS(1) if there exists a third channel C with input alphabet $\mathcal{Y}_1$ and output $\mathcal{Y}_2$ such that BMS(2) can be regarded as the concatenation of BMS(1) and C. By this we mean that passing a bit through BMS(1) and then feeding the output to C is statistically equivalent to passing the bit through BMS(2). If the transition matrix of C is $\{R(y_2|y_1)\}$, this can be written in formulae as

$$Q_2(y_2|x) = \sum_{y_1 \in \mathcal{Y}_1} R(y_2|y_1)\, Q_1(y_1|x)\,, \tag{15.21}$$

where, to simplify the notation, we assumed $\mathcal{Y}_1$ to be discrete. A pictorial representation of this relationship is provided by Fig. 15.2. A formal way of expressing the same idea is that there exists a Markov chain $X \to Y_1 \to Y_2$.

Whenever BMS(2) is physically degraded with respect to BMS(1) we shall write BMS(1) $\preceq$ BMS(2) (which is read as: BMS(1) is 'less noisy than' BMS(2)). Physical degradation is a partial ordering: If BMS(1) $\preceq$ BMS(2) and BMS(2) $\preceq$ BMS(3), then BMS(1) $\preceq$ BMS(3). Furthermore, if BMS(1) $\preceq$ BMS(2) and BMS(2) $\preceq$ BMS(1), then BMS(1) = BMS(2). However, given two binary memoryless symmetric channels, they are not necessarily ordered by physical degradation (i.e. it can happen that neither BMS(1) $\preceq$ BMS(2) nor BMS(2) $\preceq$ BMS(1)).

Here are a few examples of channel pairs ordered by physical degradation.

Example 15.7 Let $\epsilon_1, \epsilon_2 \in [0,1]$ with $\epsilon_1 \leq \epsilon_2$. Then the corresponding erasure channels are ordered by physical degradation: $\mathrm{BEC}(\epsilon_1) \preceq \mathrm{BEC}(\epsilon_2)$.

Consider in fact a channel C that has input and output alphabet $\mathcal{Y} = \{\mathtt{0}, \mathtt{1}, *\}$ (the symbol $*$ representing an erasure). On inputs $\mathtt{0}$, $\mathtt{1}$, it transmits the input unchanged with probability $1-x$ and erases it with probability x. On input $*$ it outputs an erasure. If we concatenate this channel at the output of $\mathrm{BEC}(\epsilon_1)$, we obtain a channel $\mathrm{BEC}(\epsilon)$, with $\epsilon = 1-(1-x)(1-\epsilon)$ (the probability that a bit *is not* erased is the product of the probability that it is not erased by each of the component channels). The claim is thus proved by taking $x = (\epsilon_2 - \epsilon_1)/(1-\epsilon_1)$.

Exercise 15.4 If $p_1, p_2 \in [0, 1/2]$ with $p_1 \leq p_2$, then $\mathrm{BSC}(p_1) \preceq \mathrm{BSC}(p_2)$. This can be proved by showing that $\mathrm{BSC}(p_2)$ is equivalent to the concatenation of $\mathrm{BSC}(p_1)$ with a second binary symmetric channel $\mathrm{BSC}(x)$. What value of the crossover probability x should one take?

Exercise 15.5 If $\sigma_1^2, \sigma_2^2 \in [0, \infty[$ with $\sigma_1^2 \leq \sigma_2^2$, show that $\mathrm{BAWGN}(\sigma_1^2) \preceq \mathrm{BAWGN}(\sigma_2^2)$.

If $\mathrm{BMS}(1) \preceq \mathrm{BMS}(2)$, most measures of the channel 'reliability' are ordered accordingly. Let us discuss here two important such measures: (1) conditional entropy and (2) bit error rate.

(1): Let Y_1 and Y_2 be the outputs of passing a uniformly random bit, respectively, through channels BMS(1) and BMS(2). Then $H(X|Y_1) \leq H(X|Y_2)$ (the uncertainty on the transmitted bit is larger for the 'noisier' channel). This follows immediately from the fact that $X \to Y_1 \to Y_2$ is a Markov chain by applying the data processing inequality, cf. Sec. 1.4.

(2) Assume the outputs of channels BMS(1), BMS(2), are y_1 and y_2. The MAP decision rule for x knowing y_a is $\widehat{x}_a(y_a) = \arg\max_x \mathbb{P}\{X = x | Y_a = y_a\}$, with $a = 1, 2$. The corresponding bit error rate is $\mathrm{P_b}^{(a)} = \mathbb{P}\{\widehat{x}_a(y_a) \neq x\}$. Let us show that $\mathrm{P_b}^{(1)} \leq \mathrm{P_b}^{(2)}$. As $\mathrm{BMS}(1) \preceq \mathrm{BMS}(2)$, there is a channel C such that BMS(1) concatenated with C is equivalent to BMS(2). Then $\mathrm{P_b}^{(2)}$ can be regarded as the bit error rate for a non-MAP decision rule given y_1. The rule is: transmit y_1 through C, denote by y_2 the output, and then compute $\widehat{x}_2(y_2)$. This non-MAP decision rule cannot be better than the MAP rule applied directly to y_1.

Since symmetric random variables can be associated with BMS channels (see Proposition 15.1), the notion of physical degradation of channels can be extended to symmetric random variables. Let Z_1, Z_2 be two symmetric random variables and BMS(1), BMS(2) the associated BMS channels, constructed as in the proof of proposition 15.1. We say that Z_2 is physically degraded with respect to Z_1 (and we write $Z_1 \preceq Z_2$) if BMS(2) is physically degraded with respect to BMS(1). It can be proved

that this definition is in fact independent of the choice of BMS(1), BMS(2) within the family of BMS channels associated to Z_1, Z_2.

The interesting result is that BP decoding behaves in the intuitively most natural way with respect to physical degradation. As above, we fix a particular LDPC code and look at BP message as random variables due to the randomness in the received vector $\underline{y}$.

Proposition 15.8 *Consider communication over a BMS channel using an LDPC code under the all-zero codeword assumption, and BP decoding with standard initial condition $X = 0$. If $\mathsf{B}_{i,r}(F)$ is a tree, then $h_i^{(0)} \succeq h_i^{(1)} \succeq \cdots \succeq h_i^{(t-1)} \succeq h_i^{(t)}$ for any $t \le r-1$. Analogously, if $\mathsf{B}_{i\to a,r}(F)$ is a tree, then $h_{i\to a}^{(0)} \succeq h_{i\to a}^{(1)} \succeq \cdots \succeq h_{i\to a}^{(t-1)} \succeq h_{i\to a}^{(t)}$ for any $t \le r-1$.*

We shall not prove this proposition in full generality here, but rather prove its most useful consequence for our purpose, namely the fact that the bit error rate is monotonously decreasing with t.

Proof: Under the all-zero codeword assumption, the bit error rate is $\mathbb{P}\{\widehat{x}_i^{(t)} = \mathtt{1}\} = \mathbb{P}\{h_i^{(t)} < 0\}$ (for the sake of simplicity we neglect here the case $h_i^{(t)} = 0$). Assume $\mathsf{B}_{i,r}(F)$ to be a tree and fix $t \le r-1$. Then we want to show that $\mathbb{P}\{h_i^{(t)} < 0\} \le \mathbb{P}\{h_i^{(t-1)} < 0\}$. The BP log-likelihood after T iterations on the original graph, $h_i^{(t)}$, is equal to the actual (MAP) log-likelihood for the reduced model defined on the tree $\mathsf{B}_{i,t+1}(F)$. More precisely, let us call $\mathfrak{C}_{i,t}$ the LDPC code associated to the factor graph $\mathsf{B}_{i,t+1}(F)$, and imagine the following process. A uniformly random codeword in $\mathfrak{C}_{i,t}$ is transmitted through the BMS channel yielding output $\underline{y}_t$. Define the log-likelihood ratio for bit x_i

$$\widehat{h}_i^{(t)} = \frac{1}{2}\log\left\{\frac{\mathbb{P}(x_i = \mathtt{0}|\underline{y}_t)}{\mathbb{P}(x_i = \mathtt{1}|\underline{y}_t)}\right\}, \tag{15.22}$$

and denote the MAP estimate for x_i as $\widehat{x}_i$. Clearly, $\mathbb{P}\{\widehat{x}_i = \mathtt{1}|x_i = \mathtt{0}\} = \mathbb{P}\{h_i^{(t)} < 0\}$.

Instead of this MAP decoding one can imagine to scratch all the received symbols at distance t from i, and then perform MAP decoding on the reduced information. Call $\widehat{x}_i'$ the resulting estimate. The vector of non-erased symbols is $\underline{y}_{t-1}$. The corresponding log-likelihood is clearly the BP log-likelihood after $t-1$ iterations. Therefore $\mathbb{P}\{\widehat{x}_i' = \mathtt{1}|x_i = \mathtt{0}\} = \mathbb{P}\{h_i^{(t-1)} < 0\}$. By optimality of the MAP decision rule $\mathbb{P}\{\widehat{x}_i \neq x_i\} \le \mathbb{P}\{\widehat{x}_i' \neq x_i\}$, which proves our claim. □

In the case of random LDPC codes $\mathsf{B}_{i,r}(F)$ is a tree with high probability for any fixed r, in the large block length limit. Therefore Proposition 15.8 has an immediate consequence in the asymptotic setting.

Proposition 15.9 *The density evolution random variables are ordered by physical degradation. Namely, $h^{(0)} \succeq h^{(1)} \succeq \cdots \succeq h^{(t-1)} \succeq h^{(t)} \succeq \cdots$. Analogously $h_*^{(0)} \succeq h_*^{(1)} \succeq \cdots \succeq h_*^{(t-1)} \succeq h_*^{(t)} \succeq \cdots$. As a consequence, the asymptotic bit error rate after a fixed number t of iterations $\mathrm{P}_{\mathrm{b}}^{(t)} \equiv \lim_{N\to\infty} \mathrm{P}_{\mathrm{b}}^{(N,t)}$ is monotonically decreasing with t.*

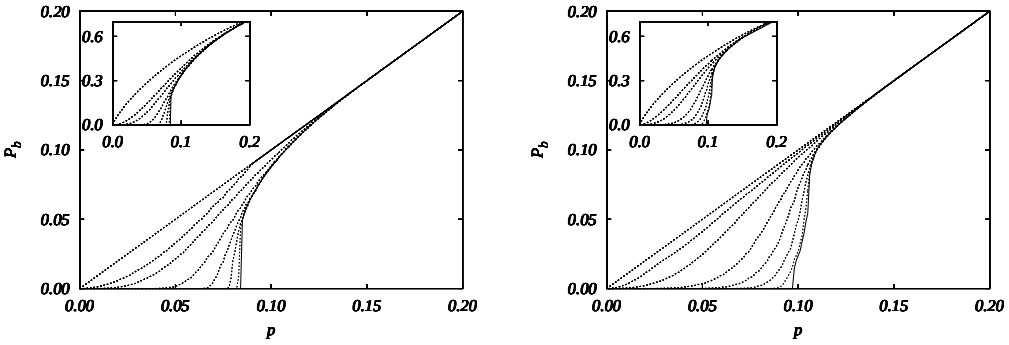

Fig. 15.3 Predicted performances of two LDPC ensembles on a BSC channel. The curves have been obtained through a numerical solution of density evolution, using population dynamics algorithm with population size $5 \cdot 10^5$. Left: the $(3, 6)$ regular ensemble. Right: an optimized irregular ensemble with the same design rate $R_{\rm des} = 1/2$, and degree distribution pair $\Lambda(x) = 0.4871\, x^2 + 0.3128\, x^3 + 0.0421\, x^4 + 0.1580\, x^{10}$, $P(x) = 0.6797\, x^7 + 0.3203\, x^8$. Dotted curves give the bit error rate obtained after $t = 1$, 2, 3, 6, 11, 21, 51 iterations (from top to bottom), and bold continuous lines to the limit $t \to \infty$. In the inset we plot the expected conditional entropy $\mathbb{E}H(X_i|\underline{Y})$.

Exercise 15.6 An alternative measure of the reliability of $h_i^{(t)}$ is provided by the conditional entropy. Assuming that a uniformly random codeword is transmitted, this is given by $H_i(t) = H(X_i|h_i^{(t)})$.

(*a*) Prove that, if $\mathsf{B}_{i,r}(F)$ is a tree, then $H_i(t)$ is monotonically decreasing with t for $t \le r - 1$.

(*b*) Assume that, under the all-zero codeword assumption $h_i^{(t)}$ has density $\mathsf{p}_t(.)$. Show that $H_i(t) = \int \log(1 + e^{-2z})\, \mathrm{d}\mathsf{p}_t(z)$. [Hint: remember that $\mathsf{p}_t(.)$ is a symmetric distribution.]

15.2.4 Numerical implementation and threshold

Density evolution is a useful tool because it can be simulated efficiently. One can estimate numerically the distributions of the density evolution variables $\{h^{(t)}, u^{(t)}\}$, as well as $\{h_*^{(t)}\}$. As we have seen this gives access to the properties of BP decoding in the large block-length limit, such as the bit error rate $\mathrm{P}_{\rm b}^{(t)}$ after t iterations.

A possible approach[1] consists in representing the distributions by samples of some fixed size S. This leads to the population dynamics algorithm discussed in Sec. 14.6.4. The algorithm generates at each time $t \in \{0, \dots, T\}$ two populations $\{h_1^{(t)}, \cdots, h_{N_{\rm pop}}^{(t)}\}$

[1] An alternative approach is as follows. Both maps (15.11) can be regarded as convolutions of probability densities for an appropriate choice of the message variables. The first one is immediate in terms of log-likelihoods. For the second map, one can use variables $r^{(t)} = (\mathrm{sign}\, h^{(t)}, \log|\tanh h^{(t)}|)$, $s^{(t)} = (\mathrm{sign}\, u^{(t)}, \log|\tanh y^{(t)}|)$). By using fast Fourier transform to implement convolutions, this can result in a significant speedup of the calculation.

l	k	$R_{\rm des}$	$p_{\rm d}$	Shannon limit
3	4	1/4	0.1669(2)	0.2145018
3	5	2/5	0.1138(2)	0.1461024
3	6	1/2	0.0840(2)	0.1100279
4	6	1/3	0.1169(2)	0.1739524

Table 15.1 Belief propagation thresholds for the BSC channel, for a few regular LDPC ensembles. The third column is the design rate $1 - l/k$.

and $\{u_1^{(t)}, \cdots, u_{N_{\rm pop}}^{(t)}\}$ which are approximately i.i.d. variables distributed respectively as $h^{(t)}$ and $u^{(t)}$. From these populations one can estimate the bit error rate following Eq. (15.13). More precisely, the population dynamics estimate is

$$\mathrm{P}_{\rm b}^{(t),\,\rm pop\,dyn} = \frac{1}{R}\sum_{n=1}^{R} \varphi\Big(B_n + \sum_{j=1}^{l(n)} u_{i_n(j)}^{(t-1)}\Big) \tag{15.23}$$

where $\varphi(x) \equiv 1$ if $x > 0$, $\varphi(0) = 1/2$, and $\varphi(x) = 0$ otherwise. Here the B_n are distributed as $\frac{1}{2}\log\frac{Q(y|0)}{Q(y|1)}$, $l(n)$ is distributed as Λ_l, and the indices $i_n(1), \dots, i_n(l)$ are uniformly random in $\{1, \dots, N_{\rm pop}\}$. The parameter R is usually taken to be of the same order as the population size.

In Fig. 15.3 we report the results of population dynamics computations for two different LDPC ensembles used on a BSC channel with crossover probability p. We consider two performance measures: the bit error rate $\mathrm{P}_{\rm b}^{(t)}$ and the conditional entropy $H^{(t)}$, which can also be easily estimated from the population.

As follows from proposition 15.9, $\mathrm{P}_{\rm b}^{(t)}$ and $H^{(t)}$ are monotonically decreasing functions of the number of iterations. One can also show that they are monotonically increasing functions of p. Since $\mathrm{P}_{\rm b}^{(t)}$ is non-negative and decreasing in t, it has a finite limit $\mathrm{P}_{\rm b}^{\rm BP} \equiv \lim_{t\to\infty} \mathrm{P}_{\rm b}^{(t)}$, which is itself non-decreasing in p (the limit curve $\mathrm{P}_{\rm b}^{\rm BP}$ is estimated in Fig. 15.3 by choosing t large enough so that $\mathrm{P}_{\rm b}^{(t)}$ is independent of t within the numerical accuracy). One defines the **BP threshold** as

$$p_{\rm d} \equiv \sup\big\{\, p \in [0, 1/2] \,:\, \mathrm{P}_{\rm b}^{\rm BP}(p) = 0 \,\big\}\;. \tag{15.24}$$

Here the subscript d stands for 'dynamical:' its intrinsic meaning and its relation with phase transitions in other combinatorial problems will be discussed in Chapter **??**. Analogous definitions can be provided for other channel families such as the erasure BEC(ϵ) or Gaussian BAWGN(σ^2) channels. In general, the definition (15.24) can be extended to any family of BMS channels BMS(p) indexed by a real parameter p which orders the channels with respect to physical degradation.

Numerical simulation of density evolution allows to determine the BP threshold $p_{\rm d}$ with good accuracy. In Table 15.2.4 we report the results of a few such results. Let us stress that the threshold $p_{\rm d}$ has an important practical meaning. For any $p < p_{\rm d}$ one can achieve arbitrarily small bit error rate with high probability by just picking one random code from the ensemble $\mathrm{LDPC}_N(\Lambda, P)$ with large N and decoding it using

BP with a large enough (but independent of N) number of iterations. For $p > p_{\rm d}$ the bit error rate is asymptotically lower bounded by $\mathrm{P}_{\rm b}^{\rm BP}(p) > 0$ for any fixed number of iterations (in practice it turns out that doing more iterations, say N^a, does not help). The value of $p_{\rm d}$ is therefore a primary measure of the performance of a code.

One possible approach to the design of good LDPC consists in sticking to random ensembles, and optimizing the degree distribution. For instance one can look for the degree distribution pair (Λ, P) with the largest BP threshold $p_{\rm BP}$, given a certain design rate $R_{\rm des} = 1 - P'(1)/\Lambda'(1)$. In the simple case of communication over the BEC, the optimization over the degree distributions can be carried out analytically, as we shall see in Sec. 15.3. For general BMS channels, it can be done numerically. One computes the threshold noise level for a given degree distribution pair using density evolution, and maximizes it by a local search procedure. Figure 15.3 shows the example of an optimized irregular ensemble with rate $1/2$ for the BSC, including variable nodes of degrees $2, 3, 4$ and 10 and check nodes of degree 7 and 8. Its threshold is $p_{\rm d} \approx 0.097$ (while Shannon's limit is 0.110).

Note that this ensemble has a finite fraction of variable nodes of degree 2. We can use the analysis in Chapter 11 to compute its weight enumerator function. It turns out that the parameter of A in Eq. (11.23) is positive. This optimized ensemble has a large number of codewords with small weight. It is surprising, and not very intuitive, that a code such that there exists codewords at sublinear distance from the transmitted one, has nevertheless a large BP threshold $p_{\rm d}$. It turns out that this phenomenon is pretty general: code ensembles that approach Shannon capacity turn out to have bad 'short distance properties'. In particular the weight enumerator exponent, discussed in Section 11.2, is positive for all values of the normalized weight. Low-weight codewords don't spoil the performance in terms of $p_{\rm d}$. They are not harmless though: they degrade the code performances at moderate block-length N, below the threshold $p_{\rm d}$. Further they prevent the block error probability from vanishing as N goes to infinity (in each codeword a fraction $1/N$ of the bits is decoded incorrectly). This phenomenon is referred to as the **error floor**.

Exercise 15.7 While the BP threshold (15.24) was defined in terms of the bit error rate, any other 'reasonable' measure of error on the decoding of a single bit would give the same result. This can be shown as follows. Let Z be a symmetric random variable and $\mathrm{P}_{\rm b} \equiv \mathbb{P}\{Z < 0\} + \frac{1}{2}\mathbb{P}\{Z = 0\}$. Show that, for any $\Delta > 0$, $\mathbb{P}\{Z < \Delta\} \le (2 + e^{2\Delta})\mathrm{P}_{\rm b}$.

Consider then a sequence of symmetric random variables $\{Z^{(t)}\}$, such that the sequence of $\mathrm{P}_{\rm b}^{(t)} \to 0$ defined as before goes to 0. Show that the distribution of $Z^{(t)}$ becomes a Dirac delta at plus infinity as $t \to \infty$.

15.2.5 Local stability

Beside numerical computation, it is useful to derive simple analytical bounds on the BP threshold. A particularly interesting bound is provided by a local stability analysis. It applies to any BMS channel, and the result depends on the specific channel only through its Bhattacharya parameter $\mathfrak{B} \equiv \sum_y \sqrt{Q(y|0)Q(y|1)} \le 1$. This parameter, that we already encountered in Ch. 11, is a measure of the channel noise level. It

preserves the ordering by physical degradation (i.e. the Bhattacharya parameters of two channels BMS(1) $\preceq$ BMS(2) satisfy $\mathfrak{B}(1) \leq \mathfrak{B}(2)$), as can be checked by explicit computation.

The local stability condition depends on the LDPC code through the fraction of vertices with degree 2, $\Lambda_2 = \lambda'(0)$, and the value of $\rho'(1) = \frac{\sum_k P_k k(k-1)}{\sum_k P_k k}$. It is expressed as:

Theorem 15.10 *Consider communication of the all-zero codeword over a binary memoryless symmetric channel with Bhattacharyia parameter $\mathfrak{B}$, using random elements from the ensemble* $\mathrm{LDPC}_N(\Lambda, P)$ *and belief propagation decoding in which the initial messages $u^{(0)}_{a\to i}$ are i.i.d. copies of a symmetric random variable. Let $\mathrm{P}_{\mathrm{b}}^{(t,N)}$ be the bit error rate after t iterations, and $\mathrm{P}_{\mathrm{b}}^{(t)} = \lim_{N\to\infty} \mathrm{P}_{\mathrm{b}}^{(t,N)}$.*

1. *If $\lambda'(0)\rho'(1)\mathfrak{B} < 1$, then there exists $\xi > 0$ such that, if $\mathrm{P}_{\mathrm{b}}^{(t)} < \xi$ for some ξ, then $\mathrm{P}_{\mathrm{b}}^{(t)} \to 0$ as $t \to \infty$.*
2. *If $\lambda'(0)\rho'(1)\mathfrak{B} > 1$, then there exists $\xi > 0$ such that $\mathrm{P}_{\mathrm{b}}^{(t)} > \xi$ for any t.*

Corollary 15.11 *Define the* ***local stability threshold*** p_{loc} *as*

$$p_{\mathrm{loc}} = \inf\left\{\, p \;\mid\; \lambda'(0)\rho'(1)\mathfrak{B}(p) > 1\right\}. \tag{15.25}$$

The BP threshold p_{BP} for decoding a communication over an ordered channel family $\mathrm{BMS}(p)$ *using random codes from the* $\mathrm{LDPC}_N(\Lambda, P)$ *ensemble satisfies:*

$$p_{\mathrm{d}} \leq p_{\mathrm{loc}}\,.$$

We shall not give the full proof of the theorem, but will explain the stability argument that underlies it. If the minimum variable node degree is 2 or larger, the density evolution recursions (15.11) have as a fixed point $h, u \stackrel{\mathrm{d}}{=} Z_\infty$, where Z_∞ is the random variable that takes value $+\infty$ with probability 1. The BP threshold p_{d} is the largest value of the channel parameter such that $\{h^{(t)}, u^{(t)}\}$ converge to this fixed point as $t \to \infty$. It is then quite natural to ask what happens if the density evolution recursion is initiated with some random initial condition that is 'close enough' to Z_∞. To this end, we consider the initial condition

$$X = \begin{cases} 0 & \text{with probability } \epsilon, \\ +\infty & \text{with probability } 1-\epsilon. \end{cases} \tag{15.26}$$

This is nothing but the log-likelihood distribution for a bit revealed through a binary erasure channel, with erasure probability ϵ.

Let us now apply the density evolution recursions (15.11) with initial condition $u^{(0)} \stackrel{\mathrm{d}}{=} X$. At the first step we have $h^{(1)} \stackrel{\mathrm{d}}{=} B + \sum_{b=1}^{l-1} X_b$, where $\{X_b\}$ are i.i.d. copies

of X. Therefore $h^{(1)} = +\infty$ unless $X_1 = \cdots = X_{l-1} = 0$, in which case $h^{(1)} \stackrel{\mathrm{d}}{=} B$. We have therefore

$$\text{With probability } \lambda_l \ : h^{(1)} = \begin{cases} B & \text{with prob. } \epsilon^{l-1}, \\ +\infty & \text{with prob. } 1-\epsilon^{l-1}. \end{cases} \tag{15.27}$$

where B is distributed as the channel log-likelihood. Since we are interested in the behavior 'close' to the fixed point Z_∞, we linearize in ϵ, thus getting

$$h^{(1)} = \begin{cases} B & \text{with prob. } \lambda_2\epsilon + O(\epsilon^2), \\ +\infty & \text{with prob. } 1-\lambda_2\epsilon + O(\epsilon^2), \\ \cdots & \text{with prob. } O(\epsilon^2). \end{cases} \tag{15.28}$$

The last line is absent here, but it will become necessary at next iterations. It signals that $h^{(1)}$ could take some other value with a negligible probability.

Next consider the first iteration at check node side: $u^{(1)} = \operatorname{atanh}\{\prod_{j=1}^{k-1} \tanh h_j^{(1)}\}$. At first order in ϵ, we need to consider only two cases. Either $h_1^{(1)} = \cdots = h_{k-1}^{(1)} = +\infty$ (this happens with probability $1-(k-1)\lambda_2\epsilon + O(\epsilon^2)$), or one of the log-likelihoods is distributed like B (with probability $(k-1)\lambda_2\epsilon + O(\epsilon^2)$). Averaging over the distribution of k, we get

$$u^{(1)} = \begin{cases} B & \text{with prob. } \lambda_2\rho'(1)\epsilon + O(\epsilon^2), \\ +\infty & \text{with prob. } 1-\lambda_2\rho'(1)\epsilon + O(\epsilon^2), \\ \cdots & \text{with prob. } O(\epsilon^2). \end{cases} \tag{15.29}$$

Repeating the argument t times (and recalling that $\lambda_2 = \lambda'(0)$), we get

$$h^{(t)} = \begin{cases} B_1 + \cdots + B_t & \text{with prob. } (\lambda'(0)\rho'(1))^t\epsilon + O(\epsilon^2), \\ +\infty & \text{with prob. } 1-(\lambda'(0)\rho'(1))^t\epsilon + O(\epsilon^2), \\ \cdots & \text{with prob. } O(\epsilon^2). \end{cases} \tag{15.30}$$

The bit error rate vanishes if and only $\mathrm{P}(t;\epsilon) = \mathbb{P}\{h^{(t)} \le 0\}$ goes to 0 as $t \to \infty$. The above calculation shows that

$$\mathrm{P}(t;\epsilon) = (\lambda'(0)\rho'(1))^t \, \epsilon \, \mathbb{P}\{B_1 + \cdots + B_t \le 0\} + O(\epsilon^2)\,. \tag{15.31}$$

The probability of $B_1 + \cdots + B_t \le 0$ is computed, to leading exponential order, using the large deviations estimates of Sec. **??**. In particular, we saw in Exercise **??** that:

$$\mathbb{P}\{B_1 + \cdots + B_t \le 0\} \doteq \left\{ \inf_{z \ge 0} \mathbb{E}\,[e^{-zB}] \right\}^t . \tag{15.32}$$

We leave to the reader the exercise of showing that, since B is a symmetric random variable, $\mathbb{E}\, e^{-zB}$ is minimized for $z = 1$, thus yielding

$$\mathbb{P}\{B_1 + \cdots + B_t \le 0\} \doteq \mathfrak{B}^t\,. \tag{15.33}$$

As a consequence, the order ϵ coefficient In Eq. (15.31) behaves, to leading exponential order, as $(\lambda'(0)\rho'(1)\mathfrak{B})^t$. Depending whether $\lambda'(0)\rho'(1)\mathfrak{B} < 1$ or $\lambda'(0)\rho'(1)\mathfrak{B} > 1$, density evolution converges or not to the error-free fixed point if initiated sufficiently close to it. The full proof relies on these ideas, but it requires to control the terms of higher order in ϵ, and other initial conditions as well.

15.3 BP decoding of the erasure channel

We now focus on the erasure channel BEC(ϵ). The analysis can be greatly simplified in this case: the BP decoding algorithm has a simple interpretation, and the density evolution equations can be studied analytically. This allows to construct **capacity achieving** ensembles, i.e. codes which are, in the large N limit, error free up to a noise level given by Shannon's threshold.

15.3.1 BP, peeling and stopping sets

We consider BP decoding, with initial condition $u^{(0)}_{a\to i} = 0$. As can be seen from Eq. (15.7), the channel log likelihood B_i can take three values: $+\infty$ (if a `0` has been received at position i), $-\infty$ (if a `1` has been received at position i), 0 (if an erasure occurred at position i).

It follows from the update equations (15.8) that the messages exchanged at any subsequent time take values in $\{-\infty, 0, +\infty\}$ as well. Consider first the equation at check nodes. If one of the incoming messages $h^{(t)}_{j\to a}$ is 0, then $u^{(t)}_{a\to i} = 0$ as well. If on the other hand $h^{(t)}_{j\to a} = \pm\infty$ for all incoming messages, then $u^{(t)}_{a\to i} = \pm\infty$ (the sign being the product of the incoming signs). Next consider the update equation at variable nodes. If $u^{(t)}_{b\to i} = 0$ for all the incoming messages, and $B_i = 0$ as well, then of course $h^{(t+1)}_{i\to a} = 0$. If on the other hand some of the incoming messages, or the received value B_i take value $\pm\infty$, then $h^{(t+1)}_{i\to a}$ takes the same value. Notice that there can never be contradicting messages (i.e. both $+\infty$ and $-\infty$) incoming at a variable node.

Exercise 15.8 Show that, if contradicting messages were sent to the same variable node, this would imply that the transmitted message was not a codeword.

The meaning of the three possible messages $\pm\infty$ and 0, and of the update equations is very clear in this case. Each time the message $h^{(t)}_{i\to a}$, or $u^{(t)}_{a\to i}$ is $+\infty$ (respectively, $-\infty$), this means that the bit x_i is `0` (respectively `1`) in all codewords that coincide with the channel output on the non-erased positions: the value of x_i is perfectly known. Vice-versa, if, $h^{(t)}_{i\to a} = 0$ (or $u^{(t)}_{a\to i} = 0$) the bit x_i is currently considered equally likely to be `0` or `1`.

The algorithm is very simple: each message changes value at most one time, either from 0 to $+\infty$, or from 0 to $-\infty$.

Exercise 15.9 To show this, consider the first time, t_1 at which a message $h^{(t)}_{i\to a}$ changes from $+\infty$ to 0. Find out what has happened at time $t_1 - 1$.

Therefore a fixed point is reached after a number of updates smaller or equal to the number of edges $N\Lambda'(1)$. There is also a clear stopping criterion: if in one update round no progress is made (i.e. if $h^{(t)}_{i\to a} = h^{(t+1)}_{i\to a}$ for all directed edges $i \to a$) then no progress will be made at successive rounds.

An alternative decoding formulation of BP decoding is the so-called **peeling algorithm**. The idea is to view decoding as a linear algebra problem. The code is defined through a linear system over $\mathbb{Z}_2$, of the form $\mathbb{H}\underline{x} = \underline{0}$. The output of an erasure channel fixes a fraction of the bits in the vector $\underline{x}$ (the non-erased ones). One is left with an inhomogeneous linear system $\mathcal{L}$ over the remaining erased bits. Decoding amounts to using this new linear system to determine the bits erased by the channel. If an equation in $\mathcal{L}$ contains a single variable x_i with non vanishing coefficient, it can be used to determine x_i, and replace it everywhere. One can then repeat this operation recursively until either all the variables have been fixed (in which case decoding is successful), or the residual linear systems includes only equations over two or more variables (in which case the decoder gets stuck).

Exercise 15.10 An explicit characterization of the fixed points of the peeling algorithm can be given in terms of **stopping sets** (or **2-cores**). **stopping set**—**see2-core** A stopping set is a subset of variable nodes in the factor graph such that each check has a number of neighbors in the subset which is either zero, or at least 2. Let S be the subset of undetermined bits when the peeling algorithm stops.

(*a*) Show that S is a stopping set.

(*b*) Show that the union of two stopping sets is a stopping set. Deduce that, given a subset of variable nodes U, there exists a unique 'largest' stopping set contained in U that contains any other stopping set in U.

(*c*) Let U be the set of erased bits. Show that S is the largest stopping set contained in U.

Exercise 15.11 Let us prove that the peeling algorithm is indeed equivalent to BP decoding. As in the previous exercise, we denote by S the largest stopping set contained in the erased set U.

(*a*) Prove that, for any edge (i, a) with $i \in S$, $u^{(t)}_{a\to i} = h^{(t)}_{a\to i} = 0$ at all times.

(*b*) Vice-versa, let S' be the set of bits that are undetermined by BP after a fixed point is reached. Show that S' is a stopping set.

(*c*) Deduce that $S' = S$ (use the maximality property of S).

15.3.2 Density evolution

Let us study BP decoding of an $\mathrm{LDPC}_N(\Lambda, P)$ code after communication through a binary erasure channel. Under the assumption that the all-zero codeword has been transmitted, messages will take values in $\{0, +\infty\}$, and their distribution can be parameterized by a single real number. We denote by z_t the probability that $h^{(t)} = 0$, and by $\hat{z}_t$ the probability that $u^{(t)} = 0$. The density evolution recursions (15.11) translate into the following recursion on $\{z_t, \hat{z}_t\}$:

$$z_{t+1} = \epsilon\lambda(\hat{z}_t)\,, \qquad \hat{z}_t = 1 - \rho(1 - z_t)\,. \tag{15.34}$$

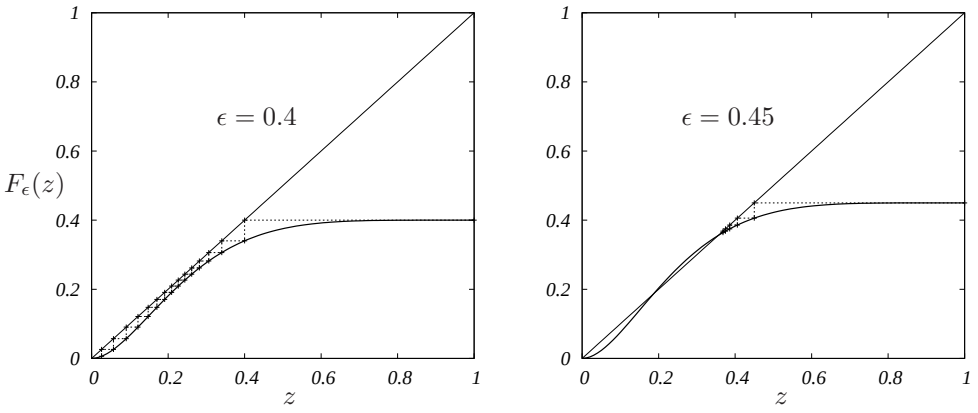

Fig. 15.4 Density evolution for the $(3, 6)$ LDPC ensemble over the erasure channel BEC(ϵ), for two values of ϵ below and above the BP threshold $\epsilon_{\rm d} = 0.4294$.

We can eliminate $\hat{z}_t$ from this recursion to get $z_{t+1} = F_\epsilon(z_t)$, where we defined $F_\epsilon(z) \equiv \epsilon\lambda(1-\rho(1-z))$. The bit error rate after t iterations in the large block-length limit is $\mathrm{P}_{\rm b}^{(t)} = \epsilon\Lambda(\hat{z}_t)$.

In Fig. 15.4 we show as an illustration the recursion $z_{t+1} = F_\epsilon(z_t)$ for the $(3, 6)$ regular ensemble. The edge perspective degree distributions are $\lambda(z) = z^2$ and $\rho(z) = z^5$, so that $F_\epsilon(z) = \epsilon[1-(1-z)^2]^5$. Notice that $F_\epsilon(z)$ is a monotonously increasing function with $F_\epsilon(0) = 0$ (if the minimum variable node degree is at least 2), and $F_\epsilon(1) = \epsilon < 1$. As a consequence the sequence $\{z_t\}$ is decreasing and converges at large t to the largest fixed point of F_ϵ. In particular $z_t \to 0$ (and consequently $\mathrm{P}_{\rm b}^{\rm BP} = 0$) if and only if $F_\epsilon(z) < z$ for all $z \in]0, 1]$. This yields the following explicit characterization of the BP threshold:

$$\epsilon_{\rm d} = \inf\left\{ \frac{z}{\lambda(1-\rho(1-z))} \; : \quad z \in]0,1] \right\} . \tag{15.35}$$

It is instructive to compare this characterization with the local stability threshold that in this case reads $\epsilon_{\rm loc} = 1/\lambda'(0)\rho'(1)$. It is obvious that $\epsilon_{\rm d} \le \epsilon_{\rm loc}$, since $\epsilon_{\rm loc} = \lim_{z\to 0} z/\lambda(1-\rho(1-z))$.

Two cases are possible, as illustrated in Fig. 15.5: either $\epsilon_{\rm d} = \epsilon_{\rm loc}$ or $\epsilon_{\rm d} < \epsilon_{\rm loc}$. Each one corresponds to a different behavior of the bit error rate. If $\epsilon_{\rm d} = \epsilon_{\rm loc}$, then, generically[2], $\mathrm{P}_{\rm b}^{\rm BP}(\epsilon)$ is a continuous function of ϵ at $\epsilon_{\rm d}$ with $\mathrm{P}_{\rm b}^{\rm BP}(\epsilon_{\rm d}+\delta) = C\delta + O(\delta^2)$ just above threshold. If on the other hand $\epsilon_{\rm d} < \epsilon_{\rm loc}$, then $\mathrm{P}_{\rm b}^{\rm BP}(\epsilon)$ is discontinuous at $\epsilon_{\rm d}$ with $\mathrm{P}_{\rm b}^{\rm BP}(\epsilon_{\rm d}+\delta) = \mathrm{P}_{\rm b}^{\rm BP,*} + C\delta^{1/2} + O(\delta)$ just above threshold.

[2]Other behaviors are possible but they are not 'robust' with respect to a perturbation of the degree sequences.

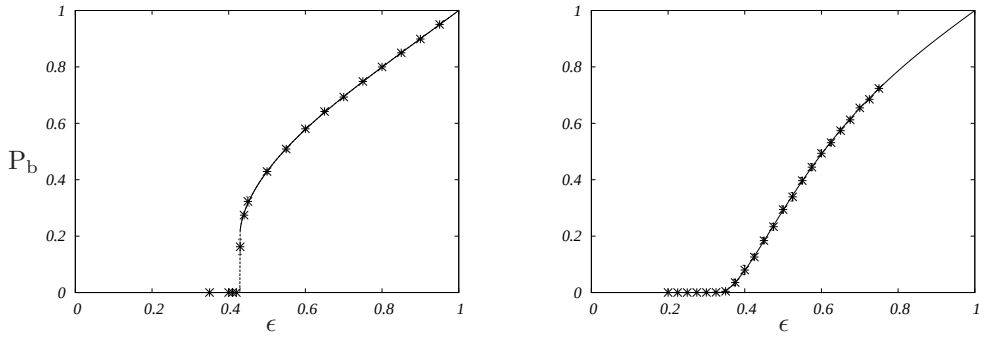

Fig. 15.5 The bit error rate under belief propagation decoding for the $(3,6)$ (left) and $(2,4)$ (right) ensembles. The prediction of density evolution (bold lines) is compared to numerical simulations (averaged over 10 code/channel realizations with block-length $N = 10^4$). For the $(3,6)$ ensemble $\epsilon_{\rm BP} \approx 0.4294 < \epsilon_{\rm loc} = \infty$, the transition is discontinuous. For the $(2,4)$ ensemble $\epsilon_{\rm BP} = \epsilon_{\rm loc} = 1/4$, the transition is continuous.

Exercise 15.12 Consider communication over the binary erasure channel using random elements from the regular (l,k) ensemble, in the limit $k, l \to \infty$, with a fixed rate $R_{\rm des} = 1 - l/k$. Prove that the BP threshold $\epsilon_{\rm d}$ tends to 0 in this limit.

15.3.3 Ensemble optimization

The explicit characterization (15.35) of the BP threshold for the binary erasure channel opens the way to the optimization of the code ensemble.

A possible setup is the following. Fix an erasure probability $\epsilon \in]0,1[$: this is the estimated noise level on the channel that we are going to use. For a given degree sequence pair (λ,ρ), let $\epsilon_{\rm d}(\lambda,\rho)$ denote the corresponding BP threshold, and $R(\lambda,\rho) = 1 - \frac{\sum_k \rho_k/k}{\sum_l \lambda_l/l}$ be the design rate. Our objective is to maximize the rate, while keeping $\epsilon_{\rm d}(\lambda,\rho) \le \epsilon$. Let us assume that the check node degree distribution ρ is given. Finding the optimal variable node degree distribution can then be recast as a (infinite dimensional) linear programming problem:

$$\begin{cases} \text{maximize} & \sum_l \lambda_l/l\,, \\ \text{subject to} & \sum_l \lambda_l = 1 \\ & \lambda_l \ge 0 \quad \forall\, l\,, \\ & \epsilon\lambda(1-\rho(1-z)) \le z \quad \forall\, z \in]0,1]\,. \end{cases} \tag{15.36}$$

Notice that the constraint $\epsilon\lambda(1-\rho(1-z)) \le z$ is conflicting with the requirement of maximizing $\sum_l \lambda_l/l$, since both are increasing functions in each of the variables λ_l. As usual with linear programming, one can show that the objective function is maximized when the constraints are satisfied with equality i.e. $\epsilon\lambda(1-\rho(1-z)) = z$ for all $z \in 0,1]$. This 'matching condition' allows to derive λ, for a given ρ.

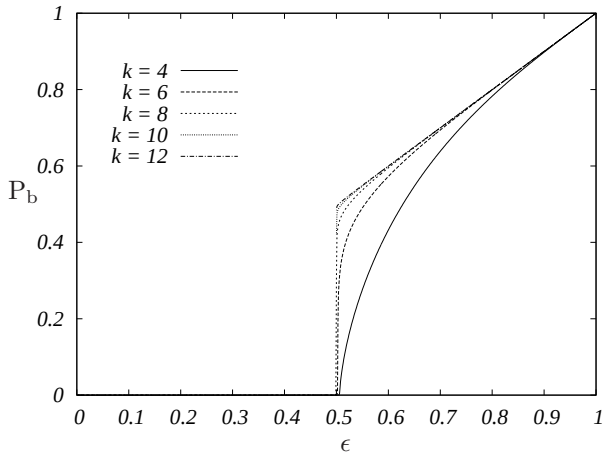

Fig. 15.6 Belief propagation bit error rate for $\mathrm{LDPC}_N(\Lambda, P)$ ensembles from the capacity achieving sequence $(\lambda^{(k)}, \rho^{(k)})$ defined in the main text. The sequence is constructed in such a way as to achieve capacity at the noise level $\epsilon = 0.5$ (the corresponding capacity is $C(\epsilon) = 1 - \epsilon = 0.5$). The 5 ensembles considered here have design rates $R_{\mathrm{des}} = 0.42253$, 0.48097, 0.49594, 0.49894, 0.49976 (respectively for $k = 4$, 6, 8, 10, 12).

We shall do this in the simple case where the check nodes have uniform degree k, i.e. $\rho(z) = z^{k-1}$. The saturation condition implies $\lambda(z) = \frac{1}{\epsilon}[1 - (1-z)^{\frac{1}{k-1}}]$. By Taylor expanding this expression we get, for $l \geq 2$

$$\lambda_l = \frac{(-1)^l}{\epsilon} \, \frac{\Gamma\left(\frac{1}{k-1} + 1\right)}{\Gamma(l)\,\Gamma\left(\frac{1}{k-1} - l + 2\right)} \, . \tag{15.37}$$

In particular $\lambda_2 = \frac{1}{(k-1)\epsilon}$, $\lambda_3 = \frac{(k-2)}{2(k-1)^2\epsilon}$, and $\lambda_l \simeq \lambda_\infty l^{-k/(k-1)}$ as $l \to \infty$. Unhappily this degree sequence does not satisfy the normalization condition in (15.36). In fact $\sum_l \lambda_l = \lambda(1) = 1/\epsilon$. This problem can however be overcome by truncating the series and letting $k \to \infty$, as shown in the exercise below. The final result is that a sequence of LDPC ensembles can be found that allows for reliable communication under BP decoding, at a rate that asymptotically achieved the channel capacity $C(\epsilon) = 1 - \epsilon$. This is stated more formally below.

Theorem 15.12 *Let $\epsilon \in (0, 1)$. Then there exists a sequence of degree distribution pairs $(\lambda^{(k)}, \rho^{(k)})$, with $\rho^{(k)}(x) = x^{k-1}$, such that $\epsilon_{\mathrm{d}}(\lambda^{(k)}, \rho^{(k)}) > \epsilon$ and $R(\lambda^{(k)}, \rho^{(k)}) \to 1 - \epsilon$.*

The precise construction of the sequence $(\lambda^{(k)}, \rho^{(k)})$ is outlined in the next exercise. In Fig. 15.6 we show the BP error probability curves for this sequence of ensembles.

Exercise 15.13 Let $\rho^{(k)}(z) = z^{k-1}$, $\hat{\lambda}^{(k)}(z) = \frac{1}{\epsilon}[1-(1-z)^{1/(k-1)}]$, and $z_L = \sum_{l=2}^{L} \hat{\lambda}_l^{(k)}$. Define $L(k,\epsilon)$ as the smallest value of L such that $z_L \geq 1$. Finally, set $\lambda_l^{(k)} = \hat{\lambda}_l^{(k)}/z_{L(k,\epsilon)}$ if $l \leq L(k,\epsilon)$ and $\lambda_l^{(k)} = 0$ otherwise.

(*a*) Show that $\epsilon\lambda^{(k)}(1-\rho^{(k)}(1-z)) < z$ for all $z \in]0,1]$, and, as a consequence $\epsilon_{\rm d}(\lambda^{(k)},\rho^{(k)}) > \epsilon$. [Hint: Use the fact that the coefficients λ_l in Eq. (15.37) are non-negative and hence $\lambda^{(k)}(x) \leq \hat{\lambda}^{(k)}(z)/z_{L(k,\epsilon)}$.]

(*b*) Show that, for any sequence $l(k)$, $\hat{\lambda}_{l(k)}^{(k)} \to 0$ as $k \to \infty$. Deduce that $L(k,\epsilon) \to \infty$ and $z_{L(k,\epsilon)} \to 1$ as $k \to \infty$.

(*c*) Prove that $\lim_{k\to\infty} R(\lambda^{(k)},\rho^{(k)}) = \lim_{k\to\infty} 1-\epsilon\, z_{L(k,\epsilon)} = 1-\epsilon$.

15.4 Bethe free-energy and MAP decoding

So far we have studied the performance of $\mathrm{LDPC}_N(\Lambda,P)$ ensembles under BP message passing decoding, in the large block-length limit. Remarkably, sharp asymptotic predictions can be obtained for optimal decoding as well, and they involve the same mathematical objects, namely messages distributions. We shall focus here on symbol MAP decoding for a channel family $\{\mathrm{BMS}(p)\}$ ordered by physical degradation. As in Ch. 11, we can define a threshold $p_{\rm MAP}$ depending on the LDPC ensemble, such that MAP decoding allows to communicate reliably at all noise levels below $p_{\rm MAP}$. We shall compute $p_{\rm MAP}$ using the Bethe free-entropy. The free-entropy of our decoding problem, averaged over the received signal, is defined as $\mathbb{E}_y \log Z(y)$. Let us see how its value can be related to the properties of MAP decoding.

A crucial step to understand MAP decoding is to estimate the typical number of inputs with non-negligible probability for a given channel output. We can quantify it precisely by introducing the 'codeword entropy density' $\mathfrak{h}_N = (1/N)\,\mathbb{E}H_N(\underline{X}|\underline{Y})$, averaged over the code ensemble (throughout this section we shall use natural logarithms in the definition of the entropies, instead of logarithms in base 2). If $\mathfrak{h}_N$ is bounded away from 0 as $N \to \infty$, the typical channel output is likely to correspond to an exponential number of inputs. If on the other hand $\mathfrak{h}_N \to 0$, the correct input has to be searched among a sub-exponential number of candidates, and one may hope to be able to decode correctly. A precise relation with the error probability is provided by Fano's inequality (1.28):

Proposition 15.13 *Denote by* $\mathrm{P}_{\rm b}^N$ *the bit error probability for communication using a code of block-length* N*. Then:*

$$\mathcal{H}(\mathrm{P}_{\rm b}^N) \geq \frac{H_N(\underline{X}|\underline{Y})}{N}\,.$$

In particular, if the entropy density $H_N(\underline{X}|\underline{Y})/N$ *is bounded away from* 0*, so is* $\mathrm{P}_{\rm b}^N$*.*

Although this gives only a bound, it suggests to identify the MAP threshold as the largest noise level such that $\mathfrak{h}_N \to 0$ as $N \to \infty$. In other words, we define

$$p_{\rm c} \equiv \sup\left\{ p \,:\, \lim_{N\to\infty} \mathfrak{h}_N = 0 \right\}, \tag{15.38}$$

and conjecture that, for LDPC ensembles, the bit error rate vanishes asymptotically if $p < p_{\rm c}$, thus implying $p_{\rm MAP} = p_{\rm c}$. Hereafter we shall use $p_{\rm c}$ (or $\epsilon_{\rm c}$ for the BEC) to denote the MAP threshold. The relation between this and similar phase transitions in other combinatorial problems will be discussed in Ch. **??**.

The conditional entropy $H_N(\underline{X}|\underline{Y})$ is directly related to the free-entropy of the model defined in (15.1). More precisely we have

$$H_N(\underline{X}|\underline{Y}) = \mathbb{E}_{\underline{y}} \log Z(\underline{y}) - N \sum_y Q(y|0) \log Q(y|0)\,, \tag{15.39}$$

where $\mathbb{E}_{\underline{y}}$ denotes expectation with respect to the output vector $\underline{y}$. In order to derive this expression, we first use the entropy chain rule to write (dropping the subscript N)

$$H(\underline{X}|\underline{Y}) = H(\underline{Y}|\underline{X}) + H(\underline{X}) - H(\underline{Y})\,. \tag{15.40}$$

Since the input message is uniform over the code, $H(\underline{X}) = N \log|\mathfrak{C}|$. Further, since the channel is memoryless and symmetric, $H(\underline{Y}|\underline{X}) = \sum_i H(Y_i|X_i) = NH(Y_i|X_i = 0) = -N\sum_y Q(y|0)\log Q(y|0)$. Finally, rewriting the distribution (15.1) as

$$p(\underline{x}|\underline{y}) = \frac{|\mathfrak{C}|}{Z(\underline{y})} p(\underline{y},\underline{x})\,, \tag{15.41}$$

we can identify (by Bayes theorem) $Z(\underline{y}) = |\mathfrak{C}|\, p(\underline{y})$. The expression (15.39) follows by putting together these contributions.

The free-entropy $\mathbb{E}_y \log Z(y)$ is the non-trivial term in Eq. (15.39). For LDPC codes, in the large N limit, it is natural to compute it using the Bethe approximation of Sec. 14.2.4. Suppose $\underline{u} = \{u_{a\to i}\}$, $\underline{h} = \{h_{i\to a}\}$ is a set of messages which solves the BP equations

$$h_{i\to a} = B_i + \sum_{b\in\partial i\setminus a} u_{b\to i}\,, \qquad u_{a\to i} = \operatorname{atanh}\left\{ \prod_{j\in\partial a\setminus i} \tanh h_{j\to a} \right\}. \tag{15.42}$$

Then the corresponding Bethe free-entropy follows from Eq. (14.28):

$$\begin{aligned}\mathbb{F}(\underline{u},\underline{h}) = &- \sum_{(ia)\in E} \log\left[\sum_{x_i} \nu_{u_{a\to i}}(x_i)\nu_{h_{i\to a}}(x_i) \right] \\ &+ \sum_{i=1}^{N} \log\left[\sum_{x_i} Q(y_i|x_i) \prod_{a\in\partial i} \nu_{u_{a\to i}}(x_i) \right] + \sum_{a=1}^{M} \log\left[\sum_{\underline{x}_a} \mathbb{I}_a(\underline{x}) \prod_{i\in\partial a} \nu_{h_{i\to a}}(x_i) \right].\end{aligned} \tag{15.43}$$

where we denote by $\nu_u(x)$ the distribution of a bit x whose log-likelihood ratio is u, given by: $\nu_u(0) = 1/(1+e^{-2u})$, $\nu_u(1) = e^{-2u}/(1+e^{-2u})$.

We are interested in the expectation of this quantity with respect to the code and channel realization, in the $N \to \infty$ limit. As in Sec. 14.6.3, we assume that messages are asymptotically identically distributed, $u_{a\to i} \stackrel{\mathrm{d}}{=} u$, $h_{i\to a} \stackrel{\mathrm{d}}{=} u$, and that messages incoming in the same node along distinct edges are asymptotically independent. Under these hypotheses we get:

$$\lim_{N\to\infty} \frac{1}{N}\mathbb{E}_y\, \mathbb{F}(\underline{u},\underline{h}) = \mathrm{f}^{\mathrm{RS}}_{u,h} + \sum_y Q(y|0)\log Q(y|0)\,, \tag{15.44}$$

where the 'shifted' free-entropy density $\mathrm{f}^{\mathrm{RS}}_{u,h}$ associated with the random variables u, h is defined by:

$$\mathrm{f}^{\mathrm{RS}}_{u,h} = -\Lambda'(1)\,\mathbb{E}_{u,h}\log\left[\sum_x \nu_u(x)\nu_h(x)\right] + \mathbb{E}_{l,y,\{u_i\}}\log\left[\sum_x \frac{Q(y|x)}{Q(y,0)}\prod_{i=1}^{l}\nu_{u_i}(x)\right] - \\ + \frac{\Lambda'(1)}{P'(1)}\,\mathbb{E}_k\mathbb{E}_{\{h_i\}}\log\left[\sum_{x_1\dots x_k}\mathbb{I}\left(x_1\oplus\cdots\oplus x_k = \mathsf{0}\right)\prod_{i=1}^{k}\nu_{h_i}(x_i)\right]\,. \tag{15.45}$$

Here k and l are distributed according to P_k and Λ_l respectively, and $u_1, u_2, \dots$ (respectively $h_1, h_2, \dots$) are i.i.d.'s and distributed as u (respectively as h).

If the Bethe free-entropy is correct, the shifted Bethe free-entropy density $\mathrm{f}^{\mathrm{RS}}_{u,h}$ is equal to the codeword entropy density $\mathfrak{h}_N$. This reasonable assumption can be turned into a rigorous inequality:

Theorem 15.14 *If u, h are symmetric random variables satisfying the distributional identities $u \stackrel{\mathrm{d}}{=} \operatorname{atanh}\left\{\prod_{i=1}^{k-1}\tanh h_i\right\}$ and $h \stackrel{\mathrm{d}}{=} B + \sum_{a=1}^{l-1} u_a$, then*

$$\lim_{N\to\infty}\mathfrak{h}_N \geq \mathrm{f}^{\mathrm{RS}}_{u,h}\,. \tag{15.46}$$

It is natural to conjecture that the correct limit is obtained by optimizing the above lower bound, i.e.

$$\lim_{N\to\infty}\mathfrak{h}_N = \sup_{u,h}\,\mathrm{f}^{\mathrm{RS}}_{u,h}\,, \tag{15.47}$$

where, once again the sup is taken over the couples of symmetric random variables u, h satisfying $u \stackrel{\mathrm{d}}{=} \operatorname{atanh}\left\{\prod_{i=1}^{k-1}\tanh h_i\right\}$ and $h \stackrel{\mathrm{d}}{=} B + \sum_{a=1}^{l-1} u_a$.

This conjecture has indeed been proved in the case of communication over the binary erasure channel for a large class of LDPC ensembles (including, for instance, regular ones).

The above expression is interesting because it establishes a bridge between BP and MAP decoding. For instance, it is immediate to show that it implies $p_{\mathrm{d}} \leq p_{\mathrm{c}}$:

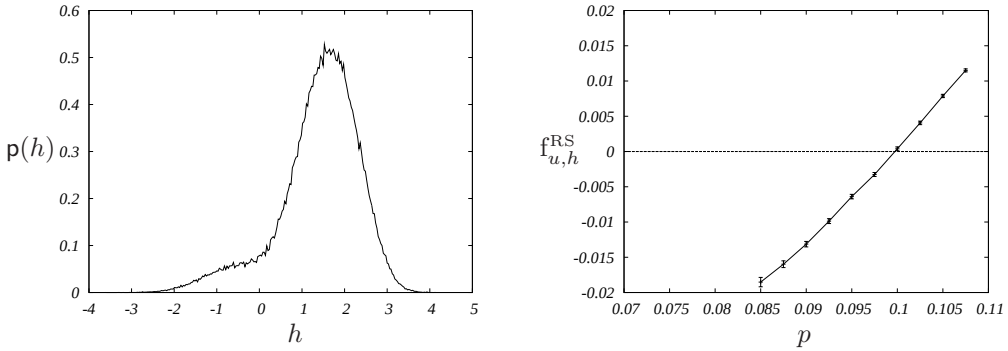

Fig. 15.7 Illustration of the RS cavity method applied to a $(3, 6)$ regular code used over the BSC channel. Left: non-trivial distribution of the h fields found by population dynamics at noise level $p = 0.095$. Right: Shifted free-entropy versus p, for the non-trivial solution (the normalization is such that the free-entropy of the perfect decoding phase $u = h = \infty$ is zero). When increasing the noise level, the non-trivial solution appears at $p_{\rm d}$, and its free-entropy becomes positive at $p_{\rm c}$

Exercise 15.14 (*a*) Recall that $u, h = +\infty$ with probability one constitute a density evolution fixed point for any noise level. Show that $\mathrm{f}^{\rm RS}_{h,u} = 0$ on such a fixed point.

(*b*) Use ordering by physical degradation to show that, if any other fixed point exists, then density evolution converges to it.

(*c*) Deduce that $p_{\rm d} \le p_{\rm c}$.

Evaluating the expression (15.47) implies an a priori infinite dimensional optimization problem. In practice good approximations can be obtained through the following procedure:

1. Initialize h, u to a couple of symmetric random variables $h^{(0)}$, $u^{(0)}$.
2. Implement numerically the density evolution recursion (15.11) by population dynamics, and iterate it until an approximate fixed point is attained.
3. Evaluate the functional $\mathrm{f}^{\rm RS}_{u,h}$ on such a fixed point, after enforcing $u \stackrel{\rm d}{=} \operatorname{atanh}\left\{\prod_{i=1}^{k-1} \tanh h_i\right\}$ exactly.

The above procedure can be repeated for several different initializations $u^{(0)}$, $h^{(0)}$. The largest of the corresponding values of $\mathrm{f}^{\rm RS}_{u,v}$ is then picked as an estimate for $\lim_{N\to\infty} \mathfrak{h}_N$.

While this procedure is not guaranteed to exhaust all the possible density evolution fixed points, it allows to compute a sequence of lower bounds to the conditional entropy density. Further, in analogy with exactly solvable cases (such as the binary erasure channel) one expects a small finite number of density evolution fixed points. In particular, for regular ensembles and $p > p_{\rm d}$, a unique (stable) fixed point is expected to exist apart from the no-error one $u, h = +\infty$. In Table 15.4 we present the

l	k	$R_{\rm des}$	$p_{\rm d}$	$p_{\rm c}$	Shannon limit
3	4	1/4	0.1669(2)	0.2101(1)	0.2145018
3	5	2/5	0.1138(2)	0.1384(1)	0.1461024
3	6	1/2	0.0840(2)	0.1010(2)	0.1100279
4	6	1/3	0.1169(2)	0.1726(1)	0.1739524

Table 15.2 MAP thresholds for the BSC channel are compared to the BP decoding thresholds, for a few regular LDPC ensembles

l	k	$R_{\rm des}$	$\epsilon_{\rm d}$	$\epsilon_{\rm c}$	Shannon limit
3	4	1/4	0.647426	0.746010	0.750000
3	5	2/5	0.517570	0.590989	0.600000
3	6	1/2	0.429440	0.488151	0.500000
4	6	1/3	0.506132	0.665656	0.666667

Table 15.3 MAP thresholds for the BEC channel are compared to the BP decoding thresholds, for a few regular LDPC ensembles

corresponding MAP thresholds for the BSC, for a few regular ensembles.

The whole approach simplifies considerably in the case of communication over the binary erasure channel, as shown in the exercise below.

Exercise 15.15 Consider the erasure channel BEC(ϵ), and look for a fixed point of the density evolution equations (15.11) such that: (*i*) $h = 0$ with probability z and $h = \infty$ with probability $1 - z$; (*ii*) $u = 0$ with probability $\hat{z}$ and $u = \infty$ with probability $1 - \hat{z}$.

(*a*) Show that z and $\hat{z}$ must satisfy the equations (15.34).

(*b*) Show that the shifted free-entropy (15.45) is equal to:

$$\mathrm{f}^{\rm RS}_{u,h} = \left[\Lambda'(1)z(1-\hat{z}) + \frac{\Lambda'(1)}{P'(1)}\left(P(1-z)-1\right) + \epsilon\Lambda(\hat{z})\right]\log 2\,. \tag{15.48}$$

(*c*) Use this expression and the conjecture (15.47) to obtain the MAP thresholds for regular ensembles of Table 15.4.

The two problems of computing the BP and MAP thresholds are thus unified by the use of the RS cavity method. For any noise level p, there always exists the solution to the RS cavity equations in which the distribution of u is a point mass distribution at $u = +\infty$, and the distribution of h is a point mass distribution at $h = +\infty$. This solution corresponds to a perfect decoding, its shifted free-entropy density is $\mathrm{f}^{\rm RS}_{u,h} = 0$. When $p > p_{\rm d}$ another solution to the RS cavity equations appears. Its shifted free-entropy density $\mathrm{f}^{\rm RS}$ can be computed from (15.44): it is initially negative and increases with p. The MAP threshold is the value $p = p_{\rm d}$ above which $\mathrm{f}^{\rm RS}$ becomes positive. Figure 15.7 illustrates this behaviour.

Still, this description leaves us with a puzzle. In the regime $p_{\rm d} \le p < p_{\rm c}$, the codeword entropy density associated to the solution $h, u < \infty$ is $\lim_{N\to\infty} \mathfrak{h}_N \ge \mathrm{f}^{\rm RS}_{u,h} < 0$. Analogously to what happens within the replica method, cf. Ch. **??**, the solution should therefore be discarded as unphysical. It turns out that a consistent picture can be obtained only by including replica symmetry breaking, which will be the object of Ch. **??**.

Notes

Belief propagation was first applied to the decoding problem by Robert Gallager in his Ph. D. thesis (Gallager, 1963), and called there 'sum-product' algorithm. Several low-complexity alternative message-passing approaches were introduced in the same work, along with the basic ideas of their analysis.

The analysis of iterative decoding of irregular ensembles over the erasure channel was pioneered by Luby and co-workers in (Luby, Mitzenmacher, Shokrollahi, Spielman and Stemann, 1997; Luby, Mitzenmacher, Shokrollahi and Spielman, 1998; Luby, Mitzenmacher, Shokrollahi and Spielman, 2001*a*; Luby, Mitzenmacher, Shokrollahi and Spielman, 2001*b*). These papers also presented the first examples of capacity achieving sequences.

Density evolution for general binary memoryless symmetric channels was introduced in (Richardson and Urbanke, 2001*b*). The whole subject is surveyed in the review (Richardson and Urbanke, 2001*a*) as well as in the upcoming book (Richardson and Urbanke, 2008). One important property we left out is 'concentration:' the error probability under message passing decoding is, for most of the codes, close to its ensemble average, that is predicted by density evolution.

The design of capacity approaching LDPC ensembles for general BMS channels is discussed in (Chung, Forney, Richardson and Urbanke, 2001; Richardson, Shokrollahi and Urbanke, 2001).

Since message passing allows for efficient decoding, one may wonder whether encoding (whose complexity is, a priori, $O(N^2)$) might become the bottleneck. Luckily this is not the case: efficient encoding schemes are discussed in (Richardson and Urbanke, 2001*c*).

The use of the RS replica method (equivalent to the cavity method) to characterize MAP decoding in sparse graph codes was initiated in (Kabashima and Saad, 1999), which considered Sourlas' LDGM codes. MN codes (a class of sparse graph codes defined by (MacKay and Neal, 1996)) and turbo codes were studied shortly after, respectively in (Kabashima, Murayama and Saad, 2000*a*; Kabashima, Murayama, Saad and Vicente, 2000*b*) and (Montanari and Sourlas, 2000; Montanari, 2000). Plain regular LDPC ensembles were considered first in (Kabashima and Saad, 2000) which considered the problem on a tree, and in (Nakamura, Kabashima and Saad, 2001). The effect of replica symmetry breaking was first investigated in (Montanari, 2001), and standard irregular ensembles were studied in (Franz, Leone, Montanari and Ricci-Tersenghi, 2002).

The fact that the RS cavity method yields the exact value of the MAP threshold and that $p_{\rm MAP} = p_{\rm c}$ has not yet been proven rigorously in a general setting. The first proof that it gives a rigorous bound was found in (Montanari, 2005), and subsequently

generalized in (Macris, 2007). An alternative proof technique uses the so-called area theorem and the related 'Maxwell construction' (Méasson, Montanari, Richardson and Urbanke, 2005*b*). Tightness of these bounds for the binary erasure channel was proved in (Méasson, Montanari and Urbanke, 2005*a*). In this case the asymptotic codeword entropy density, and the MAP threshold have been determined rigorously for a large family of ensembles.

The analysis we describe in this Chapter is valid in the large block-length limit $N \to \infty$. In practical applications, a large block-length implies some communication delay. This has motivated a number of works that aim at estimating and optimizing LDPC codes at moderate block-lengths. Some pointers to this large literature can be found in (Di, Proietti, Richardson, Telatar and Urbanke, 2002; Amraoui, Montanari, Richardson and Urbanke, 2004; Amraoui, Montanari and Urbanke, 2007; Wang, Kulkarni and Poor, 2006; Kötter and Vontobel, 2003; Stepanov, Chernyak, Chertkov and Vasic, 2005).

Appendix A
Symbols and notations

In this Appendix we summarize the conventions adopted throughout the book for symbols and notations. Secs. A.1 and A.2 deal with equivalence relations and orders of growth. Sec. A.3 presents notations used in combinatorics and probability. Table A.4 gives the main mathematical notations, and A.5 information theory notations. Table A.6 summarizes the notations used for factor graphs and graph ensembles. Table A.7 focuses on the notations used in message-passing, belief and survey propagation, and the cavity method.

A.1 Equivalence relations

As usual, the symbol $=$ denotes equality. We also use $\equiv$ for definitions and $\approx$ for 'numerically close to'. For instance we may say that the Euler-Mascheroni constant is given by

$$\gamma_{\rm E} \equiv \lim_{n\to\infty}\left(\sum_{k=1}^{n}\frac{1}{k}-\log n\right)\approx 0.5772156649\,. \tag{A.1}$$

When dealing with two random variables X and Y, we write $X \stackrel{\rm d}{=} Y$ if X and Y have the same distribution. For instance, given $n+1$ i.i.d. gaussian variables $X_0,\dots,X_n$, with zero mean and unitary variance, then

$$X_0 \stackrel{\rm d}{=} \frac{1}{\sqrt{n}}\,(X_1+\cdots+X_n)\,. \tag{A.2}$$

We adopted several equivalence symbols to denote the asymptotic behavior of functions as their argument tends to some limit. For sake of simplicity we assume here the argument to be an integer $n\to\infty$. The limit to be considered in each particular case should be clear from the context. We write $f(n)\doteq g(n)$ if f and g are equal 'to the leading exponential order' as $n\to\infty$, i.e. if

$$\lim_{n\to\infty}\frac{1}{n}\log\frac{f(n)}{g(n)}=0\,. \tag{A.3}$$

For instance we may write

$$\binom{n}{\lfloor n/2\rfloor}\doteq 2^n\,. \tag{A.4}$$

We write instead $f(n) \sim g(n)$ if f and g are asymptotically equal 'up to a constant', i.e. if

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = C\,, \tag{A.5}$$

for some constant $C \neq 0$. For instance we have

$$\frac{1}{2^n}\binom{n}{\lfloor n/2 \rfloor} \sim n^{-1/2}\,. \tag{A.6}$$

Finally, the symbol $\simeq$ is reserved for asymptoric equality, i.e. if

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 1\,. \tag{A.7}$$

For instance we have

$$\frac{1}{2^n}\binom{n}{\lfloor n/2 \rfloor} \simeq \sqrt{\frac{2}{\pi n}}\,. \tag{A.8}$$

The symbol $\cong$ denotes equality up to a constant. If $p(\,\cdot\,)$ and $q(\,\cdot\,)$ are two measures on the same finite space $\mathcal{X}$ (not necessarily normalized), we write $p(x) \cong q(x)$ if there exists $C > 0$ such that

$$p(x) = C\, q(x)\,, \tag{A.9}$$

for any $x \in \mathcal{X}$. The definition generalizes straightforwardly to infinite sets $\mathcal{X}$: the Radon-Nikodyn derivative between p and q is a positive constant.

A.2 Orders of growth

We used a couple of symbols to denote the order of growth of functions when their arguments tend to some definite limit. For sake of definiteness we refer here to functions of an integer $n \to \infty$. As above, the adaptation to any particular context should be straightforward.

We write $f(n) = \Theta(g(n))$, and say that $f(n)$ is of order $g(n)$, if there exists two positive constants C_1 and C_2 such that

$$C_1\, g(n) \leq |f(n)| \leq C_2 g(n)\,, \tag{A.10}$$

for any n large enough. For instance we have

$$\sum_{k=1}^{n} k = \Theta(n^2)\,. \tag{A.11}$$

We write instead $f(n) = o(g(n))$ if

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0\,, \tag{A.12}$$

For instance

$$\sum_{k=1}^{n} k - \frac{1}{2}\, n^2 = o(n^2)\,. \tag{A.13}$$

Finally $f(n) = O(g(n))$ if there exist a constant C such that

$$|f(n)| \leq C\, g(n) \tag{A.14}$$

for any n large enough. For instance

$$n^3 \sin(n/10) = O(n^3)\,. \tag{A.15}$$

Notice that both $f(n) = \Theta(g(n))$ and $f(n) = o(g(n))$ imply $f(n) = O(g(n))$. As the last example shows, the converse is not necessarily true.

A.3 Combinatorics and probability

The standard notation is used for multinomial coefficients. For any $n \geq 0$, $l \geq 2$ and $n_1, \ldots, n_l \geq 0$ such that $n_1 + \cdots + n_l = n$, we have:

$$\binom{n}{n_1, n_2, \ldots, n_l} \equiv \frac{n!}{n_1! n_2! \ldots n_l!}\,. \tag{A.16}$$

For binomial coefficients (i.e. for $l = 2$) the usual shorhand is

$$\binom{n}{k} \equiv \binom{n}{k, l-k} = \frac{n!}{k!(n-k)!}\,. \tag{A.17}$$

In combinatorics, certain quantities are most easily described in terms of their generating functions. Given a formal power series $f(x)$, $\mathsf{coeff}\{f(x), x^n\}$ denotes the coefficient of the monomial x^n in the series. More formally

$$f(x) = \sum_n f_n x^n \quad \Rightarrow \quad f_n = \mathsf{coeff}\{f(x), x^n\}\,. \tag{A.18}$$

For instance

$$\mathsf{coeff}\{(1+x)^m, x^n\} = \binom{m}{n}\,. \tag{A.19}$$

Some standard random variables:

- A Bernoulli p variable is a random variable X taking values in $\{0, 1\}$ such that $\mathbb{P}(X = 1) = p$.
- $B(n, p)$ denotes a binomial random variable of parameters n and p. This is defined as a random variable taking values in $\{0, \ldots, n\}$, and having probability distribution

$$\mathbb{P}\{B(n,p) = k\} = \binom{n}{k} p^k (1-p)^{n-k}\,. \tag{A.20}$$

- A Poisson random variable X of parameter λ takes integer values and has probability distribution:

$$\mathbb{P}\{X = k\} = \frac{\lambda^k}{k!} e^{-\lambda} \,. \tag{A.21}$$

The parameter λ is the mean of X.

Finally, we used the symbol δ_a for Dirac 'delta function'. This is in fact a measure, that attributes unit mass to the point a. In formulae, for any set A:

$$\delta_a(A) = \mathbb{I}(a \in A) \,. \tag{A.22}$$

A.4 Summary of mathematical notations

$=$	Equal.
$\equiv$	Defined as.
$\approx$	Numerically close to.
$\stackrel{\mathrm{d}}{=}$	Equal in distribution.
$\doteq$	Equal to the leading exponential order.
$\sim$	Asymptotically equal up to a constant.
$\cong$	Equal up to a normalization constant (for probabilities: see Eq.(14.3)).
$\Theta(f)$	Of the same order as f (see Sec. A.2).
$o(f)$	Grows more slowly than f (see Sec. A.2).
$\mathrm{argmax} f(x)$	Set of values of x where the real valued function f reaches its maximum.
$\lfloor \cdot \rfloor$	Integer part. $\lfloor x \rfloor$ is the largest integer n such that $n \le x$.
$\lceil \cdot \rceil$	$\lceil x \rceil$ is the smallest integer n such that $n \ge x$.
$\mathbb{N}$	The set of integer numbers.
$\mathbb{R}$	The set of real numbers.
$\beta \downarrow \beta_{\mathrm{c}}$	β goes to β_{c} through values $> \beta_{\mathrm{c}}$.
$\beta \uparrow \beta_{\mathrm{c}}$	β goes to β_{c} through values $< \beta_{\mathrm{c}}$.
$]a, b[$	Open interval of real numbers x such that $a < x < b$.
$]a, b]$	Interval of real numbers x such that $a < x \le b$.
$\mathbb{Z}_2$	The field of integers modulo 2.
$a \oplus b$	Sum modulo 2 of the two integers a and b.
$\mathbb{I}(\cdot)$	Indicator function: $\mathbb{I}(A) = 1$ if the logical statement A is true, $\mathbb{I}(A) = 0$ if the statement A is false .
$A \succeq 0$	The matrix A is positive semidefinite.

A.5 Information theory

H_X	Entropy of the random variable X (See Eq.(1.7)).
I_{XY}	Mutual information of the random variables X and Y (See Eq.(1.25)).
$\mathcal{H}(p)$	Entropy of a Bernoulli variable with parameter p.
$\mathfrak{M}(\mathcal{X})$	Space of probability distributions over a finite set $\mathcal{X}$.
$\mathfrak{C}$	Codebook.
$\preceq$	BMS(1) $\preceq$ BMS(2): Channel BMS(2) is physically degraded with respect to BMS(1).
$\mathfrak{B}$	Bhattacharya parameter of a channel.

A.6 Factor graphs

$\mathbb{G}_N(k, M)$	Random k-factor graph with M function nodes and N variables nodes.
$\mathbb{G}_N(k, \alpha)$	Random k-factor graph with N variables nodes. Each function node is present independently with probability $N\alpha/\binom{N}{k}$.
$\mathbb{D}_N(\Lambda, P)$	Degree constrained random factor graph ensemble.
$\mathbb{T}_r(\Lambda, P)$	Degree constrained random tree factor graph ensemble.
$\mathbb{T}_r(k, \alpha)$	Shorthand for the random tree factor graph $\mathbb{T}_r(\Lambda(x) = e^{k\alpha(x-1)}, P(x) = x^k)$.
$\Lambda(x)$	Degree profile of variable nodes.
$P(x)$	Degree profile of function nodes.
$\lambda(x)$	Edge perspective degree profile of variable nodes.
$\rho(x)$	Edge perspective degree profile of function nodes.
$\mathsf{B}_{i,r}(F)$	Neighborhood of radius r of variable node i.
$\mathsf{B}_{i\to a,t}(F)$	Directed neigborhood of an edge.

A.7 Cavity and Message passing

$\nu_{i\to a}(x_i)$	BP messages (variable to function node).
$\widehat{\nu}_{a\to i}(x_i)$	BP messages (function to variable node).
Φ	Free-entropy.
$\mathbb{F}(\underline{\nu})$	Bethe free-entropy (as a function of messages).
$\mathbb{F}^{\rm e}(\underline{\nu})$	Bethe energy (as a function of min-sum messages).
$\mathrm{f}^{\rm RS}$	Bethe (RS) free-entropy density.
$Q_{i\to a}(\nu)$	1RSB cavity message/SP message (variable to function node).
$\widehat{Q}_{a\to i}(\widehat{\nu})$	1RSB cavity message/SP message (function to variable node).
$\mathtt{x}$	Parisi 1RSB parameter.
$\mathfrak{F}(\mathtt{x})$	free-entropy density of the auxiliary model counting BP fixed points.
$\Sigma(\phi)$	Complexity.
$\mathbb{F}^{\rm RSB}(Q)$	1RSB cavity free-entropy (Bethe free-entropy of the auxiliary model, function of the messages).
$\mathrm{f}^{\rm RSB}$	1RSB cavity free-entropy density.
$\mathtt{y}$	Zero-temperature Parisi 1RSB parameter ($\mathtt{y} = \lim_{\beta\to\infty}\beta\mathtt{x}$).
$\mathfrak{F}^{\rm e}(\mathtt{y})$	Free-entropy density of the auxiliary model counting min-sum fixed points.
$\Sigma^{\rm e}(e)$	Energetic complexity.
$\mathbb{F}^{\rm RSB,e}(Q)$	Energetic 1RSB cavity free-entropy (Bethe free-entropy of the auxiliary model, function of the messages).
$\mathrm{f}^{\rm RSB,e}$	Energetic 1RSB cavity free-entropy density.

References

Abou-Chacra, R., Anderson, P. W., and Thouless, D. J. (1973). A self-consistent theory of localization. *J. Phys. C*, **6**, 1734–1752.

Achlioptas, D. (2001). Lower Bounds for Random 3-SAT via Differential Equations. *Theoretical Computer Science*, **265**, 159–185.

Achlioptas, D. (2007). Private Communication.

Achlioptas, D. and Moore, C. (2007). Random k-SAT: Two Moments Suffice to Cross a Sharp Threshold. *SIAM Journal of Computing*, **36**, 740–762.

Achlioptas, Dimitris, Naor, Assaf, and Peres, Y. (2005). Rigorous Location of Phase Transitions in Hard Optimization Problems. *Nature*, **435**, 759–764.

Achlioptas, D. and Peres, Y. (2004). The Threshold for Random k-SAT is $2k \log 2 - O(k)$. *J. Amer. Math. Soc.*, **17**, 947–973.

Aji, S.M. and McEliece, R.J. (2000). The generalized distributive law. *IEEE Trans. Inform. Theory*, **46**, 325–343.

Aldous, D. and Steele, J. M. (2003). The Objective Method: Probabilistic Combinatorial Optimization and Local Weak Convergence. In *Probability on discrete structures* (ed. H. Kesten), pp. 1–72. Springer Verlag.

Amraoui, A., Montanari, A., Richardson, T. J., and Urbanke, R. (2004). Finite-Length Scaling for Iteratively Decoded LDPC Ensembles. *IEEE Trans. Inform. Theory*.

Amraoui, A., Montanari, A., and Urbanke, R. (2007). How to Find Good Finite-Length Codes: From Art Towards Science. *Eur. Trans. on Telecomms.*, **18**, 491–508.

Applegate, D., Bixby, R., Chvátal, V., and Cook, W. The Traveling Salesman Problem. `http://www.tsp.gatech.edu/`.

Balian, R. (1992). *From Microphysics to Macrophysics: Methods and Applications of Statistical Physics*. Springer-Verlag, New York.

Barg, A. (1998). Complexity Issues in Coding Theory. In *Handbook of Coding Theory* (ed. V. S. Pless and W. C. Huffman), Chapter 7. Elsevier Science, Amsterdam.

Barg, A. and Forney, G. D. (2002). Random Codes: Minimum Distances and Error Exponents. *IEEE Trans. Inform. Theory*, **48**, 2568–2573.

Baxter, R. J. (1982). *Exactly Solved Models in Statistical Mechanics*. Academic Press, London.

Bender, E. A. and Canfield, E. R. (1978). The asymptotic number of labeled graphs with given degree sequence. *J. Comb. Theory (A)*, **24**, 296–307.

Berlekamp, E., McEliecee, R. J., and van Tilborg, H. C. A. (1978). On the inherent intractability of certain coding problems. *IEEE Trans. Inform. Theory*, **29**, 384–386.

Berrou, C. and Glavieux, A. (1996). Near optimum error correcting coding and decoding: Turbo codes. *IEEE Trans. Commun.*, **44**, 1261–1271.

Bethe, H. A. (1935). Statistical theory of superlattices. *Proc. Roy. Soc. London*

A, **150**, 552–558.
Binder, K. and Young, A. P. (1986). Spin glasses. experimental facts, theoretical concepts, and open questions. *Rev. Mod. Phys.*, **58**, 801–976.
Biroli, G. and Mézard, M. (2002). Lattice Glass Models. *Phys. Rev. Lett.*, **88**, 025501.
Bollobás, B. (1980). A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. *Eur. J. Combinatorics*, **1**, 296–307.
Bollobás, B. (2001). *Random graphs*. Cambridge University Press, Cambridge.
Bollobas, B., Borgs, C., Chayes, J. T., Kim, J. H., and Wilson, D. B. (2001). The scaling window of the 2-SAT transition. *Rand. Struct. and Alg.*, **18**, 201–256.
Bouchaud, J.-P., Cugliandolo, L., Kurchan, J., and Mézard, M. (1997). Out of equilibrium dynamics in spin glasses and other glassy systems. In *Recent progress in random magnets* (ed. A. Young). World Scientific.
Bouchaud, J.-P. and Mézard, M. (1997). Universality classes for extreme value statistics. *J. Phys. A: Math. Gen.*, **30**, 7997–8015.
Boyd, S. P. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, Cambridge.
Burshtein, D., Krivelevich, M., Litsyn, S., and Miller, G. (2002). Upper Bounds on the Rate of LDPC Codes. *IEEE Trans. Inform. Theory*, **48**, 2437–2449.
Burshtein, D. and Miller, G. (2004). Asymptotic Enumeration Methods for Analyzing LDPC Codes. *IEEE Trans. Inform. Theory*, **50**, 1115–1131.
Caracciolo, S., Parisi, G., Patarnello, S., and Sourlas, N. (1990). 3d Ising Spin Glass in a Magnetic Field and Mean-Field Theory. *Europhys. Lett.*, **11**, 783.
Chao, M.-T. and Franco, J. (1986). Probabilistic analysis of two heuristics for the 3-satisfiability problem. *SIAM J. Comput*, **15**, 1106–1118.
Chao, M. T. and Franco, J. (1990). Probabilistic analysis of a generalization of the unit-clause literal selection heuristics for the k-satisfiability problem. *Inform. Sci*, **51**, 289–314.
Chung, S.-Y., Forney, G. D., Richardson, T. J., and Urbanke, R. (2001). On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit. *IEEE Comm. Letters*, **5**, 58–60.
Chvátal, V. and Reed, B. (1992). Mick gets some (and the odds are on his side). In *Proc. of the 33rd IEEE Symposium on Foundations of Computer Science, FOCS*, Pittsburgh, pp. 620–627.
Clifford, P. (1990). Markov random fields in statistics. In *Disorder in physical systems: a volume in honour of John M. Hammersley* (ed. G. Grimmett and D. Welsh), pp. 19–32. Oxford University Press.
Cocco, S. and Monasson, R. (2001*a*). Statistical physics analysis of the computational complexity of solving random satisfiability problems using backtrack algorithms. *Eur. Phys. J. B*, **22**, 505–531.
Cocco, S. and Monasson, R. (2001*b*). Trajectories in phase diagrams, growth processes, and computational complexity: How search algorithms solve the 3-satisfiability problem. *Phys. Rev. Lett.*, **86**, 1654–1657.
Cocco, S., Monasson, R., Montanari, A., and Semerjian, G. (2006). Approximate analysis of search algorithms with physical methods. In *Computational Complexity and Statistical Physics* (ed. A. percus, G. Istrate, and C. Moore), Santa Fe Studies

in the Science of Complexity, pp. 1–37. Oxford University Press.
Conway, J. H. and Sloane, N. J. A. (1998). *Sphere Packings, Lattices and Groups*. Springer Verlag, New York.
Cook, S. A. (1971). The complexity of theorem-proving procedures. In *Proc. of the 3rd ACM Symposium on the Theory of Computing, STOC*, Shaker Heights, OH, pp. 151–158.
Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. John Wiley and sons, New York.
Csiszár, I. and Körner, J. (1981). *Information Theory: Coding Theorems for Discrete Memoryless Systems*. Academic Press, New York.
Darling, R. W. R. and Norris, J. R. (2005). Structure of large random hypergraphs. *Ann. Appl. Prob.*, **15**, 125–152.
Davis, M., Logemann, G., and Loveland, D. (1962). A machine program for theorem-proving. *Comm. ACM*, **5**, 394–397.
Davis, M. and Putnam, H. (1960). A computing procedure for quantification theory. *J. Assoc. Comput. Mach.*, **7**, 201–215.
de la Vega, W. F. (1992). On Random 2-SAT. Unpublished manuscript.
de la Vega, W. F. (2001). Random 2-SAT: results and problems. *Theor. Comput. Sci.*, **265**, 131–146.
Derrida, B. (1980). Random-energy model: Limit of a family of disordered models. *Physical Review Letters*, **45**, 79.
Derrida, B. (1981). Random-energy model: An exactly solvable model of disordered systems. *Physical Review B*, **24**, 2613–2626.
Derrida, B. and Toulouse, G. (1985). Sample to sample fluctuations in the random energy model. *J. Physique Lett.*, **46**, L223–L228.
Di, C., Montanari, A., and Urbanke, R. (2004, June). Weight Distribution of LDPC Code Ensembles: Combinatorics Meets Statistical Physics. In *Proc. of the IEEE Int. Symposium on Inform. Theory*, Chicago, USA, pp. 102.
Di, C., Proietti, D., Richardson, T. J., Telatar, E., and Urbanke, R. (2002). Finite length analysis of low-density parity-check codes on the binary erasure channel. *IEEE Trans. Inform. Theory*, **48**, 1570–1579.
Di, C., Richardson, T. J., and Urbanke, R. (2006). Weight Distribution of Low-Density Parity-Check Codes. *IEEE Trans. Inform. Theory*, **52**, 4839–4855.
Dubois, O. and Boufkhad, Y. (1997). A general upper bound for the satisfiability threshold of random r-sat formulae. *Journal of Algorithms*, **24**, 395–420.
Duchet, P. (1995). Hypergraphs. In *Handbook of Combinatorics* (ed. R. Graham, M. Grotschel, and L. Lovasz), pp. 381–432. MIT Press, Cambridge, MA.
Durrett, R. (1995). *Probability: Theory and Examples*. Duxbury Press, New York.
Edwards, S. F. and Anderson, P. W. (1975). Theory of spin glasses. *J. Phys. F*, **5**, 965–974.
Erdös, P. and Rényi, A. (1960). On the evolution of random graphs. *Publ. Math. Sci. Hung. Acad. Sci*, **5**, 17–61.
Euler, L. (1736). Solutio problematis ad geometriam situs pertinentis. *Comment. Acad. Sci. U. Petrop.*, **8**, 128–140. Reprinted in *Opera Omnia Ser.* I-7, pp. 1-10, 1766.

Feller, W. (1968). *An Introduction to Probability Theory and its Applications*. John Wiley and sons, New York.

Fischer, K. H. and Hetz, J. A. (1993). *Spin Glasses*. Cambridge University Press, Cambridge.

Flajolet, P. and Sedgewick, R. (2008). *Analytic Combinatorics*. Cambridge University Press, Cambridge.

Forney, G. D. (2001). Codes on graphs: Normal realizations. *IEEE Trans. Inform. Theory*, **47**, 520–548.

Forney, G. D. and Montanari, A. (2001). On exponential error bounds for random codes on the DMC. Available online at `http://www.stanford.edu/ montanar/PAPERS/`.

Franco, J. (2000). Some interesting research directions in satisfiability. *Annals of Mathematics and Artificial Intelligence*, **28**, 7–15.

Franz, S., Leone, M., Montanari, A., and Ricci-Tersenghi, F. (2002). Dynamic phase transition for decoding algorithms. *Phys. Rev. E*, **22**, 046120.

Franz, S. and Parisi, G. (1995). Recipes for Metastable States in Spin Glasses. *J. Physique I*, **5**, 1401.

Friedgut, E. (1999). Sharp thresholds of graph proprties, and the k-sat problem. *J. Amer. Math. Soc.*, **12**, 1017–1054.

Galavotti, G. (1999). *Statistical Mechanics: A Short Treatise*. Springer Verlag, New York.

Gallager, R. G. (1962). Low-density parity-check codes. *IEEE Trans. Inform. Theory*, **8**, 21–28.

Gallager, R. G. (1963). *Low-Density Parity-Check Codes*. MIT Press, Cambridge, Massachussetts. Available online at `http://web./gallager/www/pages/ldpc.pdf`.

Gallager, R. G. (1965). A simple derivation of the coding theorem and some applications. *IEEE Trans. Inform. Theory*, **IT-11**, 3–18.

Gallager, R. G. (1968). *Information Theory and Reliable Communication*. John Wiley and sons, New York.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York.

Garey, M. R., Johnson, D. S., and Stockmeyer, L. (1976). Some simplified NP-complete graph problems. *Theoretical Computer Science*, **1**, 237–267.

Georgii, H.-O. (1988). *Gibbs Measures and Phase Transitions*. Walter de Gruyter, Berlin - New York.

Goerdt, A. (1996). A threshold for unsatisfiability. *J. Comput. System Sci.*, **53**, 469–486.

Gomes, C. P. and Selman, B. (2005). Can get satisfaction. *Nature*, **435**, 751–752.

Gross, D. J. and Mézard, M. (1984). The simplest spin glass. *Nuclear Physics*, **B240**[**FS12**], 431–452.

Gu, J., Purdom, P. W., Franco, J., and Wah, B. W. (1996). Algorithms for the Satisfiability (SAT) Problem: A Survey. In *Satisfiability Problem:Theory and Applications* (ed. D. Du, J. Gu, and P. M. Pardalos), pp. 19–151. Amer. Math. Soc.

Guerra, F. (2005). Spin glasses. Eprint: `arXiv:cond-mat/0507581` .

Hartmann, A.K. and Weigt, M. (2005). *Phase Transitions in Combinatorial Optimization Problems*. Wiley-VCH, Weinheim, Deutschland.

Hartmann, A. K. and Rieger, H. (ed.) (2002). *Optimization Algorithms in Physics*. Wiley-VCH, Berlin.

Hartmann, A. K. and Rieger, H. (2004). *New Optimization Algorithms in Physics*. Wiley-VCH, Berlin.

Huang, K. (1987). *Statistical Mechanics*. John Wiley and sons, New York.

Janson, S., Luczak, T., and Ruciński, A. (2000). *Random graphs*. John Wiley and sons, New York.

Jaynes, E. T. (1957). Information Theory and Statistical Mechanics. *Phys. Rev.*, **106**, 620–630.

Jordan, M. (ed.) (1998). *Learning in graphical models*. MIT Press, Boston.

Kabashima, Y., Murayama, T., and Saad, D. (2000*a*). Typical Performance of Gallager-Type Error-Correcting Codes. *Phys. Rev. Lett.*, **84**, 1355–1358.

Kabashima, Y., Murayama, T., Saad, D., and Vicente, R. (2000*b*). Regular and Irregular Gallager-type Error Correcting Codes. In *Advances in Neural Information Processing Systems 12* (ed. S. A. S. et al.). MIT press, Cambridge, MA.

Kabashima, Y. and Saad, D. (1998). Belief propagation vs. TAP for decoding corrupted messages. *Europhys. Lett*, **44**, 668–674.

Kabashima, Y. and Saad, D. (1999). Statistical Mechanics of Error Correcting Codes. *Europhys. Lett.*, **45**, 97–103.

Kabashima, Y. and Saad, D. (2000). Error-correcting Code on a Cactus: a solvable Model. *Europhys. Lett.*, **51**, 698–704.

Karoński, M. and Luczak, T. (2002). The phase transition in a random hypergraph. *Journal of Computational and Applied Mathematics*, **142**, 125–135.

Katsura, S., Inawashiro, S., and Fujiki, S. (1979). Spin glasses for the infinitely long ranged bond Ising model without the use of the replica method. *Physica*, **99 A**, 193–216.

Kikuchi, R. (1951). A theory of cooperative phenomena. *Phys. Rev.*, **81**, 988–1003.

Kirkpatrick, S. and Selman, B. (1994). Critical behavior in the satisfiability of random boolean expressions. *Science*, **264**, 1297–1301.

Kirkpatrick, T. R. and Thirumalai, D. (1987). p-spin interaction spin glass models: connections with the structural glass problem. *Phys. Rev. B*, **36**, 5388–5397.

Kirkpatrick, T. R. and Wolynes, P. G. (1987). Connections between some kinetic and equilibrium theories of the glass transition. *Phys. Rev. A*, **35**, 3072–3080.

Kirousis, L. M., Kranakis, E., Krizanc, D., and Stamatiou, Y. (1998). Approximating the unsatisfiability threshold of random formulas. *Rand. Struct. and Alg.*, **12**, 253–269.

Klein, M. W., Schowalter, L. J., and Shukla, P. (1979). Spin glasses in the Bethe-Peierls-Weiss and other mean field approximations. *Phys. Rev. B*, **19**, 1492–1502.

Kötter, R. and Vontobel, P. O. (2003). Graph covers and iterative decoding of finite-length codes. In *Proc. 3rd Int. Conf. on Turbo Codes and Related Topics*, Brest, France, pp. 75–82.

Kschischang, F. R., Frey, B. J., and Loeliger, H.-A. (2001). Factor graphs and the sum-product algorithm. *IEEE Trans. Inform. Theory*, **47**, 498–519.

Lauritzen, S. L. (1996). *Graphical Models*. Oxford University Press, Oxford.

Litsyn, S. and Shevelev, V. (2003). Distance distributions in ensembles of irregular low-density parity-check codes. *IEEE Trans. Inform. Theory*, **49**, 3140–3159.

Luby, M., Mitzenmacher, M., Shokrollahi, A., and Spielman, D. A. (1998). Analysis of low density codes and improved designs using irregular graphs. In *Proc. of the 30th ACM Symposium on Theory of Computing, STOC*, pp. 249–258.

Luby, M., Mitzenmacher, M., Shokrollahi, A., and Spielman, D. A. (2001*a*). Efficient erasure correcting codes. *IEEE Trans. Inform. Theory*, **47**(2), 569–584.

Luby, M., Mitzenmacher, M., Shokrollahi, A., and Spielman, D. A. (2001*b*). Improved low-density parity-check codes using irregular graphs. *IEEE Trans. Inform. Theory*, **47**, 585–598.

Luby, M., Mitzenmacher, M., Shokrollahi, A., Spielman, D. A., and Stemann, V. (1997). Practical loss-resilient codes. In *Proc. of the 29th ACM Symposium on Theory of Computing, STOC*, pp. 150–159.

Ma, S.-K. (1985). *Statistical Mechanics*. World Scientific, Singapore.

MacKay, D. J. C. (1999). Good Error Correcting Codes Based on Very Sparse Matrices. *IEEE Trans. Inform. Theory*, **45**, 399–431.

MacKay, D. J. C. (2002). *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, Cambridge.

MacKay, D. J. C. and Neal, R. M. (1996). Near Shannon Limit Performance of Low Density Parity Check Codes. *Electronic Lett.*, **32**, 1645–1646.

Macris, N. (2007). Sharp bounds on generalised EXIT function. *IEEE Trans. Inform. Theory*, **53**, 2365–2375.

Marinari, E., Parisi, G., and Ruiz-Lorenzo, J. (1997). Numerical simulations of spin glass systems. In *Spin Glasses and Random Fields* (ed. A. Young). World Scientific.

McEliece, R. J., MacKay, D. J. C., and Cheng, J.-F. (1998). Turbo decoding as an instance of Pearl's "belief propagation" algorithm. *IEEE Jour. on Selected Areas in Communications*, **16**, 140–152.

Méasson, C., Montanari, A., Richardson, T., and Urbanke, R. (2005*b*). The Generalized Area Theorem and Some of its Consequences. submitted.

Méasson, C., Montanari, A., and Urbanke, R. (2005*a*). Maxwell Construction: The Hidden Bridge between Iterative and Maximum a Posteriori Decoding. *IEEE Trans. Inform. Theory*. accepted.

Mézard, M. and Parisi, G. (1987). Mean-Field Theory of Randomly Frustrated Systems with Finite Connectivity. *Europhys. Lett.*, **3**, 1067–1074.

Mézard, M. and Parisi, G. (1999). Thermodynamics of glasses: a first principles computation. *Phys. Rev. Lett.*, **82**, 747–751.

Mézard, M. and Parisi, G. (2001). The Bethe lattice spin glass revisited. *Eur. Phys. J. B*, **20**, 217–233.

Mézard, M., Parisi, G., and Virasoro, M. A. (1985). Random free energies in spin glasses. *J. Physique Lett.*, **46**, L217–L222.

Molloy, M. and Reed, B. (1995). A critical point for random graphs with a given degree sequence. *Rand. Struct. and Alg.*, **6**, 161–180.

Monasson, R. (1995). Structural glass transition and the entropy of metastable states. *Phys. Rev. Lett.*, **75**, 2847–2850.

Monasson, R. and Zecchina, R. (1998). Tricritical points in random combinatorics: the $(2+p)$-sat case. *J. Phys. A*, **31**, 9209–9217.

Monasson, R., Zecchina, R., Kirkpatrick, S., Selman, B., and Troyansky, L. (1999). Determining computational complexity from characteristic phase transitions. *Nature*, **400**, 133–137.

Monod, P. and Bouchiat, H. (1982). Equilibrium magnetization of a spin glass: is mean-field theory valid? *J. Physique Lett.*, **43**, 45–54.

Montanari, Andrea (2000). Turbo codes: the phase transition. *Eur. Phys. J. B*, **18**, 121–136.

Montanari, A. (2001). The glassy phase of Gallager codes. *Eur. Phys. J. B*, **23**, 121–136.

Montanari, A. (2005). Tight bounds for LDPC and LDGM codes under MAP decoding. *IEEE Trans. Inform. Theory*, **51**, 3221–3246.

Montanari, A. and Sourlas, N. (2000). The statistical mechanics of turbo codes. *Eur. Phys. J. B*, **18**, 107–119.

Mora, T. and Zdeborová, L. (2007). Random subcubes as a toy model for constraint satisfaction problems. Eprint `arXiv:0710.3804`.

Morita, T. (1979). Variational principle for the distribution function of the effective field for the random Ising model in the Bethe approximation. *Physica*, **98 A**, 566–572.

Nakamura, K., Kabashima, Y., and Saad, D. (2001). Statistical Mechanics of Low-Density Parity Check Error-Correcting Codes over Galois Fields. *Europhys. Lett.*, **56**, 610–616.

Nakanishi, K. (1981). Two- and three-spin cluster theory of spin glasses. *Phys. Rev. B*, **23**, 3514–3522.

Nishimori, H. (2001). *Statistical Physics of Spin Glasses and Information Processing*. Oxford University Press, Oxford.

Papadimitriou, C. H. (1991). On selecting a satisfying truth assignment. In *Proc. of the 32nd IEEE Symposium on Foundations of Computer Science, FOCS*, pp. 163–169.

Papadimitriou, C. H. (1994). *Computational Complexity*. Addison Wesley, Reading, MA.

Papadimitriou, C. H. and Steiglitz, K. (1998). *Combinatorial Optimization*. Dover Publications, Mineola, New York.

Parisi, G. (1988). *Statistical Field Theory*. Addison Wesley, Reading, MA.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, San Francisco.

Prim, R. C. (1957). Shortest connection networks and some generalizations. *Bell Sys. Tech. Journal*, **36**, 1389–1401.

Reif, F. (1965). *Fundamentals of statistical and thermal physics*. McGraw-Hill, New York.

Richardson, T. and Urbanke, R. (2001*a*). An introduction to the analysis of iterative coding systems. In *Codes, Systems, and Graphical Models*, IMA Volume in Mathematics and Its Applications, pp. 1–37. Springer.

Richardson, T. J., Shokrollahi, A., and Urbanke, R. (2001). Design of capacity-

approaching irregular low-density parity-check codes. *IEEE Trans. Inform. Theory*, **47**, 610–637.

Richardson, T. J. and Urbanke, R. (2001*b*). The capacity of low-density parity check codes under message-passing decoding. *IEEE Trans. Inform. Theory*, **47**, 599–618.

Richardson, T. J. and Urbanke, R. (2001*c*). Efficient encoding of low-density parity-check codes. *IEEE Trans. Inform. Theory*, **47**, 638–656.

Richardson, T. J. and Urbanke, R. (2008). *Modern Coding Theory*. Cambridge University Press, Cambridge. Available online at `http://lthcwww.epfl.ch/mct/index.php`.

Ruelle, D. (1999). *Statistical Mechanics: Rigorous Results*. World Scientific Publishing, Singapore.

Rujan, P. (1993). Finite temperature error-correcting codes. *Phys. Rev. Lett.*, **70**, 2968–2971.

Schmidt-Pruzan, J. and Shamir, E. (1985). Component structure in the evolution of random hypergraphs. *Combinatorica*, **5**, 81–94.

Schöning, U. (1999). A Probabilistic algorithm for k-SAT and constraint satisfaction problems. In *Proc. of the 40th IEEE Symposium on Foundations of Computer Science, FOCS*, New York, pp. 410–414.

Schöning, U. (2002). A Probabilistic Algorithm for k-SAT Based on Limited Local Search and Restart. *Algorithmica*, **32**, 615–623.

Selman, B. and Kautz, H. A. (1993). Domain independent extensions to gsat: Solving large structural satisfiability problems. In *Proc. IJCAI-93*, Chambery, France.

Selman, B., Kautz, H. A., and Cohen, B. (1994). Noise strategies for improving local search. In *Proc. of AAAI-94*, Seattle, WA.

Selman, B. and Kirkpatrick, S. (1996). Critical behavior in the computational cost of satisfiability testing. *Artificial Intelligence*, **81**, 273–295.

Selman, B., Mitchell, D., and Levesque, H. (1996). Generating hard satisfiability problems. *Artificial Intelligence*, **81**, 17–29.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell Sys. Tech. Journal*, **27**, 379–423, 623–655. Available electronically at `http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html`.

Sipser, M. and Spielman, D. A. (1996). Expander Codes. *IEEE Trans. Inform. Theory*, **42**, 1710–1722.

Sourlas, N. (1989). Spin-glass models as error-correcting codes. *Nature*, **339**, 693.

Spielman, D. A. (1997). The complexity of error-correcting codes. In *Lecture Notes in Computer Science 1279*, pp. 67–84.

Stepanov, M. G., Chernyak, V. Y., Chertkov, M., and Vasic, B. (2005). Diagnosis of weaknesses in modern error correction codes: a physics approach. *Phys. Rev. Lett.*, **95**, 228701–228704.

Talagrand, M. (2003). *Spin Glasses: A Challenge for Mathematicians*. Springer-Verlag, Berlin.

Tanner, R.M. (1981). A recursive approach to low complexity codes. *IEEE Trans. Inform. Theory*, **27**, 533–547.

Thouless, D. J., Anderson, P. W., and Palmer, R. G. (1977). Solution of 'Solvable model of a spin glass'. *Phil. Mag.*, **35**, 593–601.

Toulouse, G. (1977). Theory of the frustration effect in spin glasses: I. *Communications on Physics*, **2**, 115–119.

Viana, L and Bray, A. J. (1985). Phase diagrams for dilute spin glasses. *J. Phys. C*, **18**, 3037–3051.

Wainwright, M. J., Jaakkola, T. S., and Willsky, A. S. (2005*a*). A New Class of Upper Bounds on the Log Partition Function. *IEEE Trans. Inform. Theory*, **51**(7), 2313–2335.

Wainwright, M. J., Jaakkola, T. S., and Willsky, A. S. (2005*b*). MAP Estimation Via Agreement on Trees: Message-Passing and Linear Programming. *IEEE Trans. Inform. Theory*, **51**(11), 3697–3717.

Wang, C. C., Kulkarni, S. R., and Poor, H. V. (2006). Exhausting error-prone patterns in ldpc codes. *IEEE Trans. Inform. Theory*. Submitted.

Wormald, N. C. (1999). Models of random regular graphs. In *Surveys in Combinatorics, 1999* (ed. J. D. Lamb and D. A. Preece), London Mathematical Society Lecture Note Series, pp. 239–298. Cambridge University Press.

Yedidia, J. S., Freeman, W. T., and Weiss, Y. (2001). Generalized belief propagation. In *Advances in Neural Information Processing Systems, NIPS*, pp. 689–695. MIT Press.

Yedidia, J. S., Freeman, W. T., and Weiss, Y. (2005). Constructing Free Energy Approximations and Generalized Belief Propagation Algorithms. *IEEE Trans. Inform. Theory*, **51**, 2282–2313.

Yuille, A. L. (2002). CCCP Algorithms to Minimize the Bethe and Kikuchi Free Energies: Convergent Alternatives to Belief Propagation. *Neural Computation*, **14**, 691–1722.