A Modern Introduction to Bayesian Statistics

Ying Nian Wu
UCLA Department of Statistics and Data Science

Written with the help of Claude
March 2026

Contents

Preface

Statistics has two sides. One is mathematical: Bayes theorem is a consequence of the definition of conditional probability, as certain as the Pythagorean theorem and as uncontroversial. The other is philosophical: Bayesian statistics is a position on what probability means, on whether it is legitimate to assign probabilities to unknown constants, and on how prior knowledge should interact with data. These two sides are often conflated, and the conflation has generated more than two centuries of argument. This book tries to keep them clearly separate — to honor the mathematics without overselling the philosophy, and to present the philosophy without pretending it has been settled.

Striking examples, original and modern. The best way to understand an idea is to see it where it first lived. Thomas Bayes never published his central result; it was found among his papers after his death by his friend Richard Price, who recognized its significance and communicated it to the Royal Society in 1763. The problem Bayes posed — a billiard ball thrown at random, its position unknown, inferred from the outcomes of subsequent throws — is not an abstraction. It is a precise physical experiment, carefully designed so that the prior distribution is physically motivated rather than philosophically assumed. The distinction matters, and we develop it carefully.

Pierre-Simon Laplace, working independently a generation later, asked a bolder question: given that 251,527 boys and 241,945 girls were born in Paris over several decades, what is the probability that the underlying birth probability favors boys? Laplace's calculation is mathematically identical to Bayes's. But the object of inference is completely different: not a physical position, but a propensity of nature. This is the philosophical leap, and Laplace made it explicitly and unapologetically. We follow his argument step by step, because it is the clearest moment in the history of statistics where the mathematical and philosophical become distinct.

At the other end of the timeline, the same ideas reappear in the most powerful technologies of our era. Variational autoencoders, diffusion models, and large language models are all, in precise mathematical senses, Bayesian machines. The VAE maximizes an evidence lower bound that is the Bayesian free energy of statistical physics. The diffusion model's backward step is derived by applying Bayes rule to the forward noising process, and the score function that drives generation is the gradient of the log posterior. The large language model approximates a posterior predictive distribution over tokens, integrating out implicit latent structure from context. These are not

analogies. They are theorems, and we derive them.

Intuitive explanations, grounded in counting. Probability theory is, at its foundation, about counting. In the equally likely setting, $P(A) = |A|/|\Omega|$ is a ratio of counts. Bayes theorem is a column normalization in a joint table. The posterior predictive is a weighted average over candidate parameter values. Conditional independence is a statement about which columns of a table factor. Throughout this book, we return to the same concrete image: one million people — or particles, or candidates — distributed across a state space, moving according to conditional probabilities. This image is not a simplification. It is the literal meaning of probability in the finite setting, and it extends without distortion to continuous distributions, Bayes networks, MCMC samplers, and Langevin dynamics.

The Metropolis algorithm, for example, is usually introduced through detailed balance and Markov chain theory. We derive it instead by asking: if one million people are in a target distribution and each proposes to migrate to a new state, what fraction should be granted a visa to ensure the population remains in equilibrium? The answer is the Metropolis-Hastings acceptance probability, derived by counting flows. The mathematics is the same; the understanding is different.

Historical accounts. The ideas in this book did not arrive from nowhere. Bayes's result was published posthumously in 1763 by a friend who believed it answered a fundamental question about inductive reasoning. Laplace developed it into a comprehensive theory of inference, applied it to celestial mechanics, and used it to argue that the sun would rise tomorrow with overwhelming but not certain probability. The Metropolis algorithm was invented not by statisticians but by nuclear weapons physicists having recreational fun with the most powerful computer in the world, in the aftermath of the Manhattan Project. The algorithm was named after the man who controlled access to the machine, not the man who invented the method. In 2000, it was named one of the ten most important algorithms of the twentieth century. Variational inference traces to mean field theory in statistical physics, where Boltzmann distributions over interacting spin systems were approximated by factorized distributions long before the Bayesian community adopted the same idea.

These stories matter. They situate the mathematics in the human activity of trying to understand the world, and they make clear that the tools of inference are never entirely separable from the problems that produced them.

A neutral stance. This book does not advocate for Bayesian statistics as a complete philosophy of inference. We are skeptical of the classical justifications — Cox's theorem, Dutch book arguments, Savage's axioms — not because they are wrong, but because they are conditional: they tell you that *if* you want to reason consistently, or avoid being exploited, or satisfy certain behavioral axioms, then you must reason like a Bayesian. They establish the framework but say nothing about which prior to use. A practitioner who asks "what prior should I put on the speed of light?" gets no help from any of them.

Our justifications are operational rather than philosophical, and they speak directly to frequentists. The first is from decision theory: any observer who wants a scalar

criterion for comparing estimators must integrate over the parameter space with some weight function. The weight function is the prior, whether or not it is called that. The prior is not optional; it is inevitable. Making it explicit and thoughtful is simply honest. The second is from the ABC interpretation: the prior is a search strategy. It describes where to try candidate values of the unknown before filtering by the data. The posterior is the distribution of surviving candidates. This makes the prior's role concrete and constructive: it prescribes, for each specific problem, where the search should concentrate. The third is from prediction: integrating over the posterior strictly dominates plugging in any point estimate, including the MAP estimate. Full Bayesian prediction does not overfit; regularization, however well tuned, is only an approximation to this ideal.

These arguments do not require accepting any philosophical position about the meaning of probability. They require only that the observer wants to compare estimators, search the parameter space intelligently, and predict new data as accurately as possible. The prior follows from these modest requirements by arithmetic, not by epistemology.

We are equally honest about the limitations. Bayesian computation is hard: the normalizing constant is intractable for most models of practical interest, and MCMC, Langevin dynamics, and variational inference are all approximations with their own failure modes. The prior matters most when data is scarce, and specifying it requires judgment that the mathematics cannot supply. In high-dimensional models with millions of parameters, full Bayesian inference remains computationally out of reach, and most practitioners are Bayes (they use Bayes rule for latent variables) but not Bayesian (they estimate parameters by MLE or MAP rather than maintaining a posterior). We say so.

For whom this book is written. The book is written for students who have seen probability and statistics but want to understand Bayesian ideas from the ground up: where they came from, what they mean, how they are computed, and where they succeed and fail. It is also written for practitioners in machine learning who encounter Bayesian ideas in VAEs, diffusion models, and Gaussian processes and want to understand the mathematical foundations beneath them. And it is written for anyone who has wondered why a confidence interval does not mean what every non-statistician thinks it means, and what a statement that actually means what they think would look like.

The answer is in here. It requires some mathematics, some history, and a willingness to sit with genuine philosophical uncertainty rather than resolving it prematurely in either direction. We hope the journey is worth it.

Chapter 1

What is Probability?

1.1 The Observer's Uncertainty

Probability begins with an observer who does not know what is going to happen.

Roll a fair die. Before you look at the result, you are uncertain: it could be any of the six faces. Probability is a number that quantifies that uncertainty. In this simplest setting, where all outcomes are equally likely, the definition is beautifully clean.

Let $\Omega = \{1, 2, 3, 4, 5, 6\}$ be the set of all possible outcomes. Each outcome is equally likely. An **event** A is any subset of Ω — any collection of outcomes we might care about. We define:

Definition (Probability in the equally likely setting):

$$P(A) = \frac{|A|}{|\Omega|}$$

where $|A|$ denotes the number of outcomes in A and $|\Omega|$ is the total number of outcomes.

For example, the event "the die shows an even number" is $A = \{2, 4, 6\}$, which contains 3 outcomes out of 6. So $P(A) = 3/6 = 1/2$.

This is a *definition*, not a theorem. It encodes something important about the situation: there is an observer who does not yet know the outcome. Without that uncertainty, the concept of probability has no meaning. If you already know the die landed on 4, then every event either contains 4 (probability 1) or does not (probability 0). Probability lives in the gap between ignorance and knowledge.

Remark 1.1. Some textbooks define probability as the long-run frequency of an event in repeated experiments. We deliberately do not do this. The frequency interpretation is a *consequence* of the observer-uncertainty definition, as we show in the next section — it cannot serve as the foundation without circular reasoning.

1.2 The Law of Large Numbers: Frequency as a Theorem

Now suppose we roll the fair die not once, but n times. Each roll is independent, and the outcomes of all n rolls together form a **sequence** $(\omega_1, \omega_2, \ldots, \omega_n)$, where each $\omega_i \in \{1, 2, 3, 4, 5, 6\}$.

How many such sequences are there in total? For each of the n rolls, there are 6 possibilities, so the total number of sequences is 6^n. And here is the key point: since each die is fair, every one of these 6^n sequences is equally likely. We have a new sample space, Ω^n, consisting of all 6^n sequences, each with equal probability $1/6^n$.

Now consider a specific event, say $A = \{\text{the die shows a } 1\}$, which has probability $P(A) = 1/6$. In a sequence of n rolls, let $f_n(\omega_1, \ldots, \omega_n) = \frac{1}{n}\#\{i : \omega_i = 1\}$ denote the **frequency** of 1's — the proportion of rolls that landed on 1. This is a function of the sequence.

Fix a small tolerance $\epsilon > 0$, and define the set of "typical" sequences:

$$R_\epsilon = \left\{ (\omega_1, \ldots, \omega_n) \in \Omega^n : \left| f_n(\omega_1, \ldots, \omega_n) - \frac{1}{6} \right| < \epsilon \right\}$$

These are the sequences where the frequency of 1's stays within ϵ of $1/6$.

Theorem 1.2 (Law of Large Numbers)**.** $P(R_\epsilon) = |R_\epsilon|/6^n \to 1$ *as* $n \to \infty$*, for any fixed* $\epsilon > 0$*.*

In plain language: as the number of rolls grows, the overwhelming majority of sequences have the frequency of 1's very close to 1/6. An observer who does not know which sequence will occur can be nearly certain, for large n, that the frequency will be close to 1/6.

This is a counting statement. We count the number of sequences in R_ϵ and compare to the total 6^n. For large n, that fraction approaches 1. No philosophy is required — just careful counting in the sample space Ω^n.

Key insight: The frequentist interpretation of probability — that probability equals long-run frequency — is a *theorem*, not a definition. It is a logical consequence of the observer-uncertainty definition, proved by counting. Defining probability as frequency would use the conclusion to justify the premise.

The circularity of defining probability as frequency. Suppose someone insists: "The probability of rolling a 1 is 1/6 because, if I roll the die infinitely many times, exactly 1/6 of the outcomes will be 1." This definition has two problems. First, we cannot actually roll a die infinitely many times. Second, and more fundamentally, the statement "the frequency approaches 1/6" is itself a probabilistic statement — it says it is *likely* that the frequency is close to 1/6. To make this precise requires the

concept of probability we were trying to define. The argument goes in a circle. Starting from observer uncertainty, by contrast, the frequency statement is a clean theorem with no circularity.

1.3 Unequal Probabilities: Refining the Sample Space

So far, all outcomes have been equally likely. But many situations involve unequal probabilities. The key insight is that unequal probabilities can always be understood as equal probabilities on a *finer* sample space.

A fair coin from a fair die. Suppose we want to model a fair coin flip, where heads (H) and tails (T) each have probability $1/2$. We can generate this from our die: declare H if the die shows $\{1, 2, 3\}$, and T if it shows $\{4, 5, 6\}$. In the fine sample space $\Omega = \{1, 2, 3, 4, 5, 6\}$, each outcome has probability $1/6$. The event $H = \{1, 2, 3\}$ has $|H| = 3$ outcomes, so $P(H) = 3/6 = 1/2$. The coin is fair because we assigned equal-sized groups.

An unfair coin from a fair die. Now suppose we want a coin that shows heads with probability $1/3$ and tails with probability $2/3$. Simply partition the die differently: $H = \{1, 2\}$ and $T = \{3, 4, 5, 6\}$. Then $P(H) = 2/6 = 1/3$ and $P(T) = 4/6 = 2/3$. We have constructed an unfair coin from a fair die by changing the size of the groups.

More generally, if we want an event A to have probability $p = k/N$ for some integers k and N, we construct a fine sample space with N equally likely atoms and assign k of them to A. The equally likely intuition is preserved at every step. Probabilities like $1/3$, $2/5$, or $3/8$ are not fundamentally different from $1/2$ — they are just different ways of dividing a fine sample space into groups.

Remark 1.3. For continuous distributions, such as a needle that lands at a uniformly random angle between 0 and 360 degrees, the same idea applies: imagine finer and finer equally spaced grids on the sample space, and take the limit. The observer-uncertainty interpretation carries through to the continuous case, with probability now measuring the proportion of a continuous space rather than a count.

1.4 Conditional Probability: Updating the Observer's Uncertainty

Probability quantifies uncertainty before the outcome is known. But what happens when the observer learns *partial* information? The answer is **conditional probability**: a tool for updating uncertainty in light of new knowledge.

A Room of 100 People

Imagine a room containing 100 people. Of these, 50 are male and 50 are female. Among the 50 males, 40 are taller than 6 feet. Among the 50 females, 10 are taller than 6 feet.

We pick one person uniformly at random from the room.

We can lay out all the numbers in a simple table:

	Taller than 6 ft	Not taller than 6 ft	Total
Male	40	10	50
Female	10	40	50
Total	50	50	100

The sample space is $\Omega = \{\text{all 100 people}\}$, and since we pick uniformly at random, each person has probability 1/100. Define two events:

- $M =$ "the selected person is male" $\Rightarrow P(M) = 50/100 = 1/2$
- $T =$ "the selected person is taller than 6 ft" $\Rightarrow P(T) = 50/100 = 1/2$

What is $P(T \mid M)$?

Suppose we are told the selected person is male. This rules out all 50 females, restricting our attention to the 50 males. Among those 50 males, 40 are tall. So the probability of being tall, *given* the person is male, is:

$$P(T \mid M) = \frac{40}{50} = \frac{4}{5}$$

Notice what happened: we **shrank the sample space** from all 100 people to just the 50 males, and then counted within that smaller world. Formally, $|T \cap M| = 40$ (people who are both tall and male) and $|M| = 50$, so:

$$P(T \mid M) = \frac{|T \cap M|}{|M|} = \frac{40}{50} = \frac{4}{5}$$

What is $P(M \mid T)$?

Now flip the question. We are told the selected person is tall. Among the 50 tall people, how many are male? From the table, 40 of the 50 tall people are male. So:

$$P(M \mid T) = \frac{40}{50} = \frac{4}{5}$$

This time we shrank the sample space to the 50 tall people and counted the males among them:

$$P(M \mid T) = \frac{|M \cap T|}{|T|} = \frac{40}{50} = \frac{4}{5}$$

Notice that $P(T \mid M) = P(M \mid T) = 4/5$ here — a numerical coincidence of this particular example. In general, $P(A \mid B)$ and $P(B \mid A)$ are *different* quantities, and the relationship between them is precisely what Bayes rule describes.

The General Definition

In the equally likely setting, conditioning on B shrinks the sample space to B and counts outcomes in A within that smaller world. This gives:

$$P(A \mid B) = \frac{|A \cap B|}{|B|}$$

We can rewrite this using probabilities directly by dividing numerator and denominator by $|\Omega|$:

$$P(A \mid B) = \frac{|A \cap B|/|\Omega|}{|B|/|\Omega|} = \frac{P(A \cap B)}{P(B)}$$

This formula holds not just in the equally likely setting but for *any* probability model, and we adopt it as the general definition.

Definition (Conditional Probability):

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}, \qquad \text{provided } P(B) > 0$$

Conditioning on B is equivalent to treating B as the new sample space: all probability is redistributed within B, and renormalized so that $P(B \mid B) = 1$.

Computing $P(M \mid T)$ Without Counting

Let us redo the $P(M \mid T)$ calculation using the formula, without directly counting. We have:

$$P(M \cap T) = \frac{40}{100} = 0.4$$
$$P(T) = \frac{50}{100} = 0.5$$

Therefore:

$$P(M \mid T) = \frac{P(M \cap T)}{P(T)} = \frac{0.4}{0.5} = \frac{4}{5}$$

This calculation did not require us to directly count the 50 tall people and identify the males among them. We used only $P(M \cap T)$ — the overall fraction of people who are both male and tall — and $P(T)$ — the overall fraction who are tall. Dividing one by the other gives the conditional probability. This is already the essence of Bayes rule.

Remark 1.4. Notice the asymmetry in the two conditional probabilities, $P(T \mid M)$ and $P(M \mid T)$. The quantity $P(T \mid M)$ was easy to read off directly: we were *given* the breakdown of heights within each gender. The quantity $P(M \mid T)$, on the other hand,

required a calculation — we had to "invert" the conditioning. This asymmetry is not accidental. In most real problems, one direction of conditioning is given by the structure of the world (for example, a disease causes symptoms with known probabilities), and the other direction is what we want to reason about (what is the probability of disease given that we observe symptoms?). Bayes rule is the systematic tool for performing this inversion, and we develop it fully in the next chapter.

Exercise 1.5. In the same room of 100 people, suppose we now learn that the selected person is *not* tall. Compute $P(M \mid T^c)$, where T^c is the event that the person is not taller than 6 feet. Does knowing someone is short make it more or less likely they are male, compared to no information? Explain why this makes intuitive sense given the table.

Exercise 1.6. Suppose a room has 200 people: 120 males and 80 females. Among the males, 30 are taller than 6 feet. Among the females, 20 are taller than 6 feet. A person is chosen at random.

1. Construct a table like the one above.

2. Compute $P(T)$, $P(T \mid M)$, $P(T \mid F)$, and $P(M \mid T)$.

3. Verify that $P(M \mid T)$ computed from the formula $P(M \cap T)/P(T)$ agrees with direct counting.

1.5 Random Variables and Notation

So far we have worked with events — subsets of a sample space Ω — and assigned them probabilities. In practice, we rarely describe events by listing their elements explicitly. Instead, we describe them through **random variables**: quantities whose value depends on which outcome $\omega \in \Omega$ occurs. This section introduces the notation that connects events, random variables, and probability, and clears up the relationship between them once and for all.

The Sample Space of People

Return to the room of 100 people from Section 1.4. The sample space is $\Omega = \{\omega_1, \omega_2, \ldots, \omega_{100}\}$, where each ω is a person. We pick one person uniformly at random.

Each person carries attributes. For any person ω, let:

- $X(\omega)$ denote the **gender** of ω, taking values in $\{\text{male}, \text{female}\}$.

- $Y(\omega)$ denote the **height** of ω, taking values in $(0, \infty)$.

These are functions: $X : \Omega \to \{\text{male}, \text{female}\}$ and $Y : \Omega \to (0, \infty)$. Each one assigns a numerical or categorical value to every person in the population. A **random variable** is exactly this: a function defined on the sample space. We call it "random" because ω is chosen at random, so the value $X(\omega)$ is not known in advance.

Remark 1.7. The description of a random variable as a function on Ω is the formally precise definition, and it is worth understanding at least once. In everyday work, however, we rarely need to think about the underlying ω explicitly. We simply write X for the random variable and treat it as an uncertain quantity, without worrying about which function it is. The function-on-Ω picture is most useful when you want to be absolutely clear about what "two random variables defined on the same probability space" means, or when you need to reason carefully about joint distributions.

From Events to Random Variables

Any statement about X or Y defines an event — a subset of Ω. For example:

- The event "the selected person is male" is:

  $$A = \{\omega \in \Omega : X(\omega) = \text{male}\}$$

  This is the set of all people in Ω whose gender function equals male. Among our 100 people, $|A| = 50$, so $P(A) = 50/100 = 1/2$.

- The event "the selected person is taller than 6 feet" is:

  $$B = \{\omega \in \Omega : Y(\omega) > 6\}$$

  From our table, $|B| = 50$, so $P(B) = 1/2$.

- The event "the selected person is male *and* taller than 6 feet" is:

  $$A \cap B = \{\omega \in \Omega : X(\omega) = \text{male and } Y(\omega) > 6\}$$

  with $|A \cap B| = 40$, so $P(A \cap B) = 40/100 = 0.4$.

Three Ways to Write the Same Probability

There are three standard ways to write the probability of an event defined through a random variable. They all mean the same thing:

The three notations for probability:

$$\begin{aligned} P(A) &= P(\{\omega : X(\omega) = \text{male}\}) \\ &= P(X = \text{male}) \\ &= p(\text{male}) \end{aligned}$$

- $P(A)$: probability of the **event** $A \subseteq \Omega$. The most formal notation, emphasizing the set of outcomes.
- $P(X = \text{male})$: probability that the **random variable** X takes the value "male". This is the most common notation in statistics; it makes the variable and the value both explicit.
- $p(\text{male})$: the **probability mass function (pmf)** evaluated at the value "male". The lower-case p signals that we have fixed the random variable X and are treating $p(\cdot)$ as a function of the value alone.

Upper-case P takes an event or a statement about random variables as its argument. Lower-case p takes a specific value as its argument. Both refer to the same underlying probability measure on Ω.

All three notations are used throughout this book, and all three appear in the literature. The choice is usually one of convenience and clarity in context, not a difference in meaning.

Conditional Probability in the Three Notations

The conditional probability $P(A \mid B)$ from Section 1.4 can likewise be written in three equivalent ways:

$$\begin{aligned} P(A \mid B) &= P(\{\omega : X(\omega) = \text{male}\} \mid \{\omega : Y(\omega) > 6\}) \\ &= P(X = \text{male} \mid Y > 6) \\ &= p(\text{male} \mid Y > 6) \end{aligned}$$

We computed this in Section 1.4: $P(X = \text{male} \mid Y > 6) = 40/50 = 0.8$.

Distributions of Random Variables

The full collection of probabilities $p(x) = P(X = x)$ for all values x is called the **distribution** of X. For the gender variable X:

Value x	$p(x) = P(X = x)$
male	0.50
female	0.50

This is the **probability mass function (pmf)** of X: a function that assigns a probability to every possible value of a discrete random variable. The probabilities are non-negative and sum to one.

Density Functions for Continuous Variables

For a continuous random variable like height Y, individual values have probability zero: $P(Y = y) = 0$ for every specific y. This is not a paradox — it simply reflects that among one million people, the number whose height is *exactly* 5.832... feet is zero. Probability only makes sense for *intervals*.

Discretizing into bins. The natural remedy is to stop asking about exact values and instead ask about small bins. Fix a small bin width $\Delta y > 0$ and consider the event:

$$\{Y \in (y,\, y + \Delta y)\} = \{\omega \in \Omega : y < Y(\omega) \leq y + \Delta y\}$$

This event has positive probability. We define the **probability density function (pdf)** $p_Y(y)$ by writing:

$$P(Y \in (y,\, y + \Delta y)) = p_Y(y)\, \Delta y$$

So $p_Y(y)$ is not itself a probability. It is probability *per unit length*: the probability of the bin is the density times the width of the bin. Geometrically, $p_Y(y)\, \Delta y$ is the **area** of a thin rectangle of height $p_Y(y)$ and width Δy sitting above the point y on the horizontal axis. The density curve $y \mapsto p_Y(y)$ is the shape whose area gives probabilities.

Since the total probability over all bins must equal one:

$$\sum_{\text{bins}} p_Y(y)\, \Delta y = 1 \qquad \xrightarrow{\Delta y \to 0} \qquad \int_{-\infty}^{\infty} p_Y(y)\, dy = 1$$

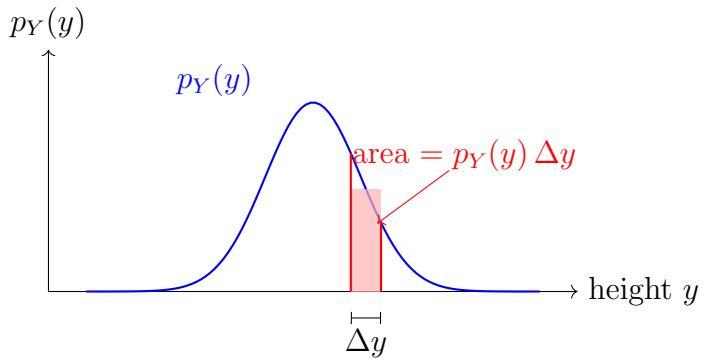
The probability that Y falls in any interval $[a, b]$ is the area under the density curve over that interval:

$$P(a \leq Y \leq b) = \int_a^b p_Y(y)\, dy$$

The one million people picture. The density has a vivid interpretation in the one million people (or particles) picture. Place all one million people on the number line, each at their height $Y(\omega)$. The result is a scatter of points on the line. The bin $(y, y + \Delta y)$ contains some number of these points. The fraction of the million people whose height falls in that bin is:

$$P(Y \in (y,\, y + \Delta y)) = \frac{\text{number of people in } (y,\, y + \Delta y)}{1{,}000{,}000} = p_Y(y)\, \Delta y$$

So $p_Y(y)$ measures the **density of people** near height y: how many people (in millions) are packed into one unit of height at y. Where the density curve is tall, people are tightly packed. Where it is low, people are sparse. The density curve is literally a picture of how the population is distributed along the height axis.



The joint density of two variables. Now suppose each person has both a height $Y(\omega)$ and a weight $Z(\omega)$. Place the million people as points in the (y, z) plane, one point per person. The result is a **scatterplot**. Some regions of the plane are densely populated (many people of similar height and weight); others are sparse.

The **joint density** $p(y, z)$ measures how densely the people are packed near the point (y, z):

$$P(Y \in (y,\ y + \Delta y),\ Z \in (z,\ z + \Delta z)) = p(y, z)\,\Delta y\,\Delta z$$

This is the fraction of the million people who fall in the small rectangle $(y, y+\Delta y) \times (z, z + \Delta z)$ — a small patch of the scatterplot. The joint density is the people-density of the scatterplot: tall where the scatter is dense, low where it is sparse.

Marginalizing: collapsing the scatterplot. To recover the marginal density $p_Y(y)$ from the joint, sum over all values of Z. In the scatterplot picture, this means collapsing all points onto the y-axis (the height axis), ignoring weight entirely:

$$p_Y(y)\,\Delta y = \sum_{\text{bins in } z} p(y, z)\,\Delta y\,\Delta z \quad \xrightarrow{\Delta z \to 0} \quad p_Y(y) = \int p(y, z)\,dz$$

Conditioning: slicing the scatterplot. The conditional density $p(z \mid y)$ is the distribution of Z among all people whose height is near y. In the scatterplot, fix a thin vertical strip at y, look at all the people in that strip, and ask how they are distributed along the z-axis. Formally:

$$p(z \mid y) = \frac{p(y, z)}{p_Y(y)}$$

The denominator $p_Y(y)$ is the density of people in the strip; dividing by it renormalizes so that the conditional density integrates to one. This is the continuous version of Rule 3 from Chapter 2: normalization by restricting to a slice.

The density picture in one sentence. A probability density $p(y)$ measures the density of people (or particles) near y in the one million people scatterplot. The probability of any region is the fraction of people in that region, which equals the

area (in 1D) or volume (in 2D) under the density surface over that region.

Remark 1.8. The density $p_Y(y)$ can be greater than 1 at some points. This is not a problem: it is a density per unit length, not a probability. A probability density of 5 at some y simply means the people are very tightly packed near y — five million people per unit of height. The *probability* over a bin is $p_Y(y)\,\Delta y$, which is always between 0 and 1 as long as Δy is small enough.

Joint, Marginal, and Conditional Distributions

The notation extends naturally to two or more random variables.

- The **joint distribution** $p(x, y) = P(X = x, Y = y)$ gives the probability that both $X = x$ and $Y = y$ simultaneously. For one discrete and one continuous variable, the joint is a density in y for each fixed value of x: $p(x, y) = P(X = x, Y \in (y, y + dy))/dy$.

- The **marginal distribution** $p(x) = P(X = x)$ is obtained by summing (or integrating) the joint over all values of Y:

$$p(x) = \sum_y p(x, y) \quad \text{(discrete } Y\text{)} \qquad \text{or} \qquad p(x) = \int p(x, y)\, dy \quad \text{(continuous } Y\text{)}$$

  This is Rule 2 (summarization) from Chapter 2, in the notation of distributions.

- The **conditional distribution** $p(y \mid x) = P(Y = y \mid X = x)$ gives the distribution of Y among all people with $X = x$. From the definition of conditional probability:

$$p(y \mid x) = \frac{p(x, y)}{p(x)}$$

  This is Rule 3 (normalization) from Chapter 2. For continuous Y, $p(y \mid x)$ is a conditional density: a density in y that integrates to one for each fixed x.

Summary of notation conventions:

- Upper-case letters X, Y, Z denote **random variables**.
- Lower-case letters x, y, z denote **specific values** (also called realizations) of those variables.
- $P(\cdot)$ takes an event or a statement as its argument: $P(X = x)$, $P(Y > 6)$, $P(X = x \mid Y = y)$.
- $p(\cdot)$ is a pmf or pdf, a function of specific values: $p(x)$, $p(y)$, $p(x, y)$, $p(y \mid x)$.

- When the random variable is clear from context, $p(x)$ is shorthand for $P(X = x)$. When it is not, write $p_X(x)$ to indicate explicitly that p is the distribution of X.

These conventions are widespread but not universal. In some texts, upper-case P is used for everything; in others, f is used for densities. We follow the lower-case p convention throughout because it treats discrete and continuous distributions uniformly, which is exactly what the one million people picture does.

Exercise 1.9. Let $\Omega = \{\omega_1, \ldots, \omega_{200}\}$ be the population from Exercise 1.2 (120 males, 80 females; 30 males taller than 6 ft, 20 females taller than 6 ft). Define $X(\omega)$ as the gender of ω and $Y(\omega)$ as the indicator that ω is taller than 6 ft ($Y = 1$ if tall, $Y = 0$ otherwise).

1. Write out the joint pmf $p(x, y)$ for all four combinations of $x \in \{\text{male}, \text{female}\}$ and $y \in \{0, 1\}$ as a 2×2 table.
2. Verify that the four entries sum to 1.
3. Compute the marginals $p_X(x)$ and $p_Y(y)$ by summing the rows and columns of the table.
4. Compute the conditional pmf $p(x \mid y = 1)$ from the formula $p(x, y)/p(y)$, and verify it matches your direct count from Exercise 1.2.

Chapter 2

The Three Rules of Probability

All of probability theory — from simple medical tests to deep generative models — rests on three rules. In this chapter we derive them from a single concrete image: one million people moving through a world of causes and effects. Everything follows from counting.

2.1 One Million People

Imagine exactly one million people. Each person is in one of three **cause states**, $\mathcal{X} = \{1, 2, 3\}$. We write $p(x)$ for the number of people in state x, measured in millions. So $p(x)$ is literally a count, expressed in units of one million people:

Cause state x	People (millions)	$p(x)$
$x = 1$	500,000	0.5
$x = 2$	300,000	0.3
$x = 3$	200,000	0.2
Total	1,000,000	1.0

Because the total is one million, the counts in millions always sum to one:

$$p(1) + p(2) + p(3) = 0.5 + 0.3 + 0.2 = 1$$

Now each person independently moves from their cause state x to one of four **effect states**, $\mathcal{Y} = \{a, b, c, d\}$. The quantity $p(y \mid x)$ is the *fraction* of people in cause state x who move to effect state y. Fractions within each cause state must sum to one:

	$y = a$	$y = b$	$y = c$	$y = d$
$x = 1$	0.2	0.3	0.4	0.1
$x = 2$	0.4	0.2	0.1	0.3
$x = 3$	0.1	0.5	0.2	0.2

For example, of the 500,000 people in state 1, a fraction 0.2 move to a, a fraction 0.3 move to b, and so on. The movement is **physical**: it happens in the world, driven by the cause, regardless of any observer.

From this picture, three rules emerge immediately by counting.

2.2 Rule 1: Factorization

How many people (in millions) are in cause state x *and* end up in effect state y? Of the $p(x)$ million people in state x, a fraction $p(y \mid x)$ move to y. Multiplying:

$$p(x, y) = p(x) \cdot p(y \mid x)$$

This is multiplication of a count by a fraction — nothing more. Let us fill in the full joint table. Each entry is $p(x) \cdot p(y \mid x)$:

	$y = a$	$y = b$	$y = c$	$y = d$	Row total
$x = 1$	$0.5 \times 0.2 = 0.10$	$0.5 \times 0.3 = 0.15$	$0.5 \times 0.4 = 0.20$	$0.5 \times 0.1 = 0.05$	0.50
$x = 2$	$0.3 \times 0.4 = 0.12$	$0.3 \times 0.2 = 0.06$	$0.3 \times 0.1 = 0.03$	$0.3 \times 0.3 = 0.09$	0.30
$x = 3$	$0.2 \times 0.1 = 0.02$	$0.2 \times 0.5 = 0.10$	$0.2 \times 0.2 = 0.04$	$0.2 \times 0.2 = 0.04$	0.20

Each entry $p(x, y)$ is the number of millions of people who are simultaneously in cause state x and effect state y.

Rule 1 — Factorization (Chain Rule):

$$p(x, y) = p(x)\, p(y \mid x)$$

The number of people in (x, y) jointly equals the number in x, times the fraction of those who move to y.

2.3 Rule 2: Summarization

How many people (in millions) end up in effect state $y = a$, regardless of which cause state they came from? We add up the contributions from all three cause states:

$$\begin{aligned} p(a) &= p(1, a) + p(2, a) + p(3, a) \\ &= 0.10 + 0.12 + 0.02 = 0.24 \end{aligned}$$

So 240,000 people end up in state a: 100,000 from state 1, 120,000 from state 2, and 20,000 from state 3. Doing this for all effect states:

	$y = a$	$y = b$	$y = c$	$y = d$
Column total $p(y)$	0.24	0.31	0.27	0.18

These four numbers sum to $0.24 + 0.31 + 0.27 + 0.18 = 1$, as they must: every person ends up somewhere.

Rule 2 — Summarization (Law of Total Probability):

$$p(y) = \sum_x p(x, y) = \sum_x p(x)\, p(y \mid x)$$

The total number of people in effect state y is the sum over all cause states of those who started in x and moved to y. We are *summing out* the variable x to get the marginal count in y.

2.4 Rule 3: Normalization and Bayes Rule

Now comes the most important question: among all the people who ended up in effect state $y = a$, what fraction came from cause state $x = 1$?

We know 240,000 people ended up in a. Of those, 100,000 came from cause state 1. So the fraction is:

$$p(1 \mid a) = \frac{100{,}000}{240{,}000} = \frac{0.10}{0.24} \approx 0.417$$

Similarly:

$$p(2 \mid a) = \frac{0.12}{0.24} = 0.500$$
$$p(3 \mid a) = \frac{0.02}{0.24} \approx 0.083$$

These three numbers sum to $0.417 + 0.500 + 0.083 = 1$: every person in state a came from somewhere. What we have just done is **normalize** within the column $y = a$.

In general, the fraction of the $p(y)$ million people in effect state y who came from cause state x is:

$$p(x \mid y) = \frac{p(x, y)}{p(y)}$$

Substituting Rule 1 into the numerator and Rule 2 into the denominator:

Rule 3 — Normalization (Bayes Rule):

$$p(x \mid y) = \frac{p(x)\, p(y \mid x)}{\sum_{x'} p(x')\, p(y \mid x')}$$

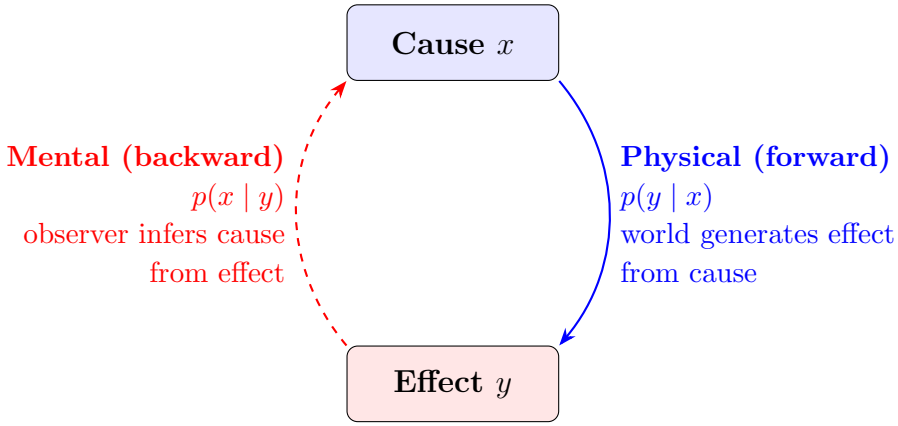
Among all people who arrived at effect state y, the fraction who came from cause state x is proportional to $p(x)$ (how many started in x) times $p(y \mid x)$ (how likely they were to move to y from x). The denominator normalizes so that the fractions sum to one.

These three rules are complete. Every probability calculation, from the simplest clinical test to the most complex generative model, is a composition of these three operations: factorize, summarize, normalize.

2.5 The Physical and Mental Directions

The one million people migrate *forward*, from cause to effect. This is the **physical direction**: it is what the world does, driven by natural processes, independent of any observer. A disease causes symptoms. A source emits a signal. A gene produces a phenotype.

The statistician's task runs in the opposite direction: given that we have observed an effect y, reason backward to the most likely cause x. This is the **mental direction**: it happens in the mind of the observer, not in the world. It requires both the physical model $p(y \mid x)$ and a prior $p(x)$ describing how the causes were distributed before the observation.



- $p(y \mid x)$: **Physical (forward).** The cause x produces the effect y. In our example, being in state x determines, via the transition fractions $p(y \mid x)$, where you migrate. This is given by the structure of the world.
- $p(x \mid y)$: **Mental (backward).** The observer sees the effect y and reasons back to the cause x. In our example, looking at the column for $y = a$ and normalizing by the column total gives the posterior fractions $p(x \mid a)$. This requires the prior $p(x)$ and proceeds via Bayes rule.

Central insight. The physical direction $p(y \mid x)$ is given by nature. The mental direction $p(x \mid y)$ is the task of the statistician. Bayes rule is the mathematically precise, logically consistent solution to reasoning in the direction opposite to causality. It is not a philosophical choice — it is the unique answer to the question: among all people who arrived at y, what fraction came from x?

2.6 Continuous Variables: Population Density

So far, the cause states x and effect states y have been discrete — a small finite list. In many real problems, x and y take continuous values: a person's height, a physical constant, the weight vector of a model. The three rules and Bayes rule continue to hold, but we need to replace counts with *densities* and sums with *integrals*. The ideas are the same; only the bookkeeping changes.

From Counts to Density

Start with something familiar. The city of Los Angeles covers approximately 1302 square kilometers and is home to about 4 million people. If we pick a small neighborhood of area ΔA square kilometers, the number of people living there is approximately:

$$\text{number of people} \approx \frac{4{,}000{,}000}{1302} \cdot \Delta A$$

The factor $4{,}000{,}000/1302 \approx 3072$ is the **population density**: the number of people per square kilometer. In general:

$$\text{population density at location } x = \frac{\text{number of people near } x}{\text{area near } x}$$

Density is a count divided by a size. It allows us to talk about how many people are in a small region, even when the region contains no meaningful integer count on its own.

Normalized Density: Probability Density

Now replace the raw count with a *proportion*. Instead of counting people, measure the fraction of the total population in each region. If the total population is N and $\Delta N(x)$ people live in the small interval $(x,\, x + \Delta x)$, define:

$$p(x)\,\Delta x = \frac{\Delta N(x)}{N}$$

The left side is the fraction of the population in the bin $(x, x + \Delta x)$, and $p(x)$ is the **probability density**: the fraction per unit length. Because every person lives

somewhere, summing over all bins gives:

$$\sum_{\text{bins}} p(x)\,\Delta x = 1$$

As the bin width $\Delta x \to 0$, this sum becomes an integral:

$$\int p(x)\,dx = 1$$

The density $p(x)$ is not itself a probability — it is a probability *per unit length*. The probability of finding a randomly chosen person in the interval $(x, x + \Delta x)$ is $p(x)\,\Delta x$, not $p(x)$ alone.

The Three Rules with Densities

Now place our one million people along a continuous cause axis $x \in \mathbb{R}$ and a continuous effect axis $y \in \mathbb{R}$. The three rules carry over exactly, with sums replaced by integrals and counts replaced by densities.

Rule 1 — Factorization. The joint density is the product of the marginal density and the conditional density. In the bin $(x, x + \Delta x)$ there are $p(x)\,\Delta x$ million people. Of those, the fraction $p(y \mid x)\,\Delta y$ move to the bin $(y, y + \Delta y)$. The number who do both is:

$$p(x, y)\,\Delta x\,\Delta y = p(x)\,\Delta x \cdot p(y \mid x)\,\Delta y$$

Canceling $\Delta x\,\Delta y$:

$$p(x, y) = p(x)\,p(y \mid x)$$

The formula is identical to the discrete case.

Rule 2 — Summarization. The marginal density $p(y)$ is obtained by summing the joint over all cause bins and taking $\Delta x \to 0$:

$$p(y)\,\Delta y = \sum_{\text{bins in } x} p(x, y)\,\Delta x\,\Delta y \quad \Rightarrow \quad p(y) = \int p(x, y)\,dx = \int p(x)\,p(y \mid x)\,dx$$

This is the continuous version of "sum over all causes to get the total arriving at effect y."

Rule 3 — Normalization (Bayes Rule). Among all people in the thin strip at effect value y — there are $p(y)\,\Delta y$ of them — what fraction came from the bin at x? The joint bin at (x, y) contains $p(x, y)\,\Delta x\,\Delta y$ people. Dividing by the strip total $p(y)\,\Delta y$:

$$p(x \mid y)\,\Delta x = \frac{p(x, y)\,\Delta x\,\Delta y}{p(y)\,\Delta y} = \frac{p(x, y)}{p(y)}\,\Delta x$$

The factors of Δy cancel before we take any limit. Dividing both sides by Δx:

$$p(x \mid y) = \frac{p(x, y)}{p(y)} = \frac{p(x)\,p(y \mid x)}{\int p(x')\,p(y \mid x')\,dx'}$$

Bayes Rule (continuous version):

$$p(x \mid y) = \frac{p(x)\, p(y \mid x)}{\displaystyle\int p(x')\, p(y \mid x')\, dx'}$$

The cancellation of Δy in the derivation is essential: it explains why conditioning on a continuous observation y is well-defined even though the probability of any single exact value $P(Y = y) = 0$. The bin sizes cancel before we take the limit, leaving a ratio of densities.

Remark 2.1. The integral $\int p(x')\, p(y \mid x')\, dx'$ in the denominator is the continuous analogue of the column total in our discrete example: it sums up the contributions from all possible cause values x' to the effect y. This integral is called the **marginal likelihood** or **evidence**, and computing it is the central computational challenge of Bayesian statistics, to which we return in later chapters.

Exercise 2.2. In the numerical example of this chapter, compute $p(x \mid b)$ for each $x \in \{1, 2, 3\}$. Verify that the three values sum to 1. Which cause state is most likely given that we observe $y = b$? Does this make intuitive sense given the transition table?

Exercise 2.3. Suppose the one million people are now in cause states $x \in \{1, 2\}$ with $p(1) = 0.7$ and $p(2) = 0.3$, and effect states $y \in \{a, b\}$ with transition fractions:

	$y = a$	$y = b$
$x = 1$	0.1	0.9
$x = 2$	0.8	0.2

1. Compute the joint table $p(x, y)$.
2. Compute the marginals $p(a)$ and $p(b)$.
3. Compute $p(x = 1 \mid y = a)$. Is it larger or smaller than the prior $p(1) = 0.7$? Explain why this makes sense in terms of the physical direction.

Exercise 2.4. Let $x \sim \text{Uniform}[0, 1]$, so $p(x) = 1$ for $x \in [0, 1]$, and let $y \mid x \sim \text{Uniform}[0, x]$, so $p(y \mid x) = 1/x$ for $0 \le y \le x$. Compute $p(x \mid y)$ for a fixed $y \in [0, 1]$. (Hint: first compute $p(x, y)$, then $p(y) = \int_y^1 p(x, y)\, dx$, then normalize.) What family of distribution does $p(x \mid y)$ belong to?

Chapter 3

Applications of Bayes Rule

3.1 Medical Diagnosis

A classic application illustrating the physical/mental distinction. Let $x \in \{0, 1\}$ denote disease status and $y \in \{0, 1\}$ denote test result.

- $p(x = 1) = 0.001$: prior prevalence (1 in 1000 people have the disease)
- $p(y = 1|x = 1) = 0.99$: sensitivity (test positive given disease) — **physical**
- $p(y = 1|x = 0) = 0.05$: false positive rate — **physical**

A patient tests positive. What is the probability they have the disease?

$$p(x = 1|y = 1) = \frac{p(x = 1)p(y = 1|x = 1)}{p(x = 1)p(y = 1|x = 1) + p(x = 0)p(y = 1|x = 0)}$$

$$= \frac{0.001 \times 0.99}{0.001 \times 0.99 + 0.999 \times 0.05} \approx \frac{0.00099}{0.05094} \approx 0.019$$

Only about 2%! This surprises most people. The one million people picture makes it transparent: of 1,000,000 people, about 1,000 have the disease and 999 test positive. Of 999,000 without the disease, about 49,950 test falsely positive. So among the $999 + 49{,}950 \approx 50{,}950$ who test positive, only 999 actually have the disease — about 2%.

The lesson: A highly accurate test applied to a rare disease yields mostly false positives. The prior prevalence $p(x)$ matters enormously. Ignoring the prior and reporting only sensitivity and specificity gives a dangerously incomplete picture.

3.2 Independence

Two events or random variables are **independent** when knowing one tells you nothing about the other. This section makes that intuition precise, first through counting in a concrete population, then through a continuous geometric example, and finally through the formal definition.

A Population of 200 People

Consider a population of 200 people: 100 males and 100 females. Among the 100 males, 20 hold a college degree. Among the 100 females, 20 also hold a college degree. The full table is:

	Degree	No degree	Total	$P(\text{degree} \mid \cdot)$
Male	20	80	100	$20/100 = 0.20$
Female	20	80	100	$20/100 = 0.20$
Total	40	160	200	$40/200 = 0.20$

Let D be the event "holds a degree" and M the event "is male." We have $P(D \mid M) = 0.20$ and $P(D \mid M^c) = 0.20$. Knowing a person's gender does not change our probability that they hold a degree. Gender and degree are **independent**.

Notice also that $P(D \cap M) = 20/200 = 0.10$, while $P(D) \cdot P(M) = 0.20 \times 0.50 = 0.10$. The joint probability factors into the product of the marginals.

Two Uniform Random Variables

For continuous random variables, independence has a clean geometric interpretation. Let $X \sim \text{Uniform}[0,1]$ and $Y \sim \text{Uniform}[0,1]$ be independent. The pair (X, Y) is then *uniformly distributed over the unit square* $[0,1]^2$: every region of the square has probability equal to its area.

Take any interval $A \subseteq [0,1]$ on the x-axis and any interval $B \subseteq [0,1]$ on the y-axis. The event $\{X \in A\}$ corresponds to a vertical strip of width $|A|$ in the square; the event $\{Y \in B\}$ corresponds to a horizontal strip of height $|B|$. Their intersection $\{X \in A, Y \in B\}$ is a rectangle of area $|A| \cdot |B|$. Therefore:

$$P(X \in A,\ Y \in B) = |A| \cdot |B| = P(X \in A) \cdot P(Y \in B)$$

The joint probability is the product of the marginal probabilities. Equivalently, the joint density factors: $p(x, y) = p(x) \cdot p(y) = 1 \cdot 1 = 1$ on the unit square. Knowing $X = x$ does not change the distribution of Y: $p(y \mid x) = p(y) = 1$.

The Formal Definition

We can now state independence precisely in both event and random variable language.

Independence of Events. Two events A and B are **independent**, written $A \perp B$, if any of the following equivalent conditions holds:

$$\begin{aligned} P(A \cap B) &= P(A)\,P(B) \\ P(A \mid B) &= P(A) \qquad (\text{provided } P(B) > 0) \\ P(B \mid A) &= P(B) \qquad (\text{provided } P(A) > 0) \end{aligned}$$

Independence of Random Variables. Two random variables X and Y are **independent**, written $X \perp Y$, if their joint density (or mass function) factors:

$$p(x, y) = p(x)\,p(y) \quad \text{for all } x, y$$

Equivalently, $p(y \mid x) = p(y)$ for all x: the conditional distribution of Y given $X = x$ is the same regardless of x.

Why the three event conditions are equivalent. Starting from $P(A \cap B) = P(A)P(B)$, divide both sides by $P(B)$ to get $P(A \mid B) = P(A)$. This says that knowing B occurred does not update the probability of A: B carries no information about A. By symmetry, knowing A also carries no information about B. All three conditions express the same thing: the two events have no influence on each other.

Remark 3.1. Independence and mutual exclusivity are easily confused but are almost opposite concepts. If A and B are mutually exclusive ($A \cap B = \emptyset$), then $P(A \cap B) = 0$. For independence we need $P(A \cap B) = P(A)P(B)$. These are equal only if $P(A) = 0$ or $P(B) = 0$. Two events with positive probability cannot be both independent and mutually exclusive: knowing that one occurred rules the other out, which is the opposite of being uninformative.

3.3 Conditional Independence

Independence is rarely exact in the raw data. Two variables may appear strongly related when we look at the whole population, yet become unrelated once we account for a third variable. Or the reverse: two variables may look unrelated overall but are in fact connected once we condition on a common context. The right concept for reasoning about these situations is **conditional independence**.

Conditional Independence. Events A and B are **conditionally independent**

given C, written $A \perp B \mid C$, if:

$$P(A \cap B \mid C) = P(A \mid C)\, P(B \mid C)$$

Equivalently, $P(A \mid B, C) = P(A \mid C)$: once we know C, learning B provides no additional information about A.

Random variables X and Y are **conditionally independent given** Z, written $X \perp Y \mid Z$, if:

$$p(x, y \mid z) = p(x \mid z)\, p(y \mid z) \quad \text{for all } x, y, z$$

Equivalently, $p(x \mid y, z) = p(x \mid z)$: knowing $Z = z$, additional knowledge of Y does not change the distribution of X.

Conditional independence is *not* the same as unconditional independence. We will see examples in both directions: variables that are unconditionally independent but conditionally dependent, and variables that are unconditionally dependent but conditionally independent.

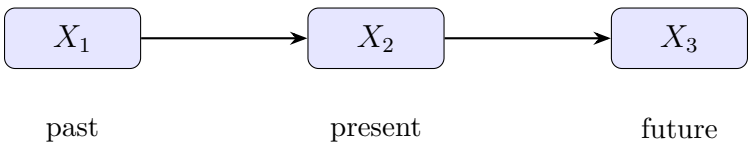
Scenario 1: The Markov Property

Consider three random variables representing a process unfolding over time: X_1 (past), X_2 (present), X_3 (future). Think of a person's location yesterday, today, and tomorrow. The **Markov property** states:

$$X_3 \perp X_1 \mid X_2 \qquad \Longleftrightarrow \qquad p(x_3 \mid x_1, x_2) = p(x_3 \mid x_2)$$

Once we know where the person is *today*, knowing where they were *yesterday* gives no additional information about where they will be *tomorrow*. The present screens off the past.

In the cause-effect language of this book: X_1 is a **remote cause** and X_2 is the **immediate cause** of X_3. All of the influence that X_1 can have on X_3 is transmitted through X_2; there is no direct channel from past to future that bypasses the present. Conditioning on the present intercepts that channel completely.



The Markov property does *not* say that X_1 and X_3 are unconditionally independent. Before we observe X_2, knowing X_1 does tell us something about X_3: the past influences the future through the present. It is only after conditioning on X_2 that the channel is blocked.

Scenario 2: Shared Cause

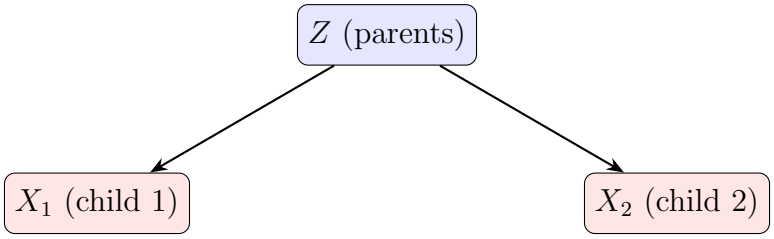
Now consider two children born to the same parents. Let Z represent the parents' genetic makeup, and X_1, X_2 the genotypes of the two children. The biological mechanism is:

- Each child inherits their genotype independently from the same parents.
- The parents' genes are the *common cause* of both children's traits.

Given the parents' genotype Z, the two children's genotypes are **conditionally independent**:

$$p(x_1, x_2 \mid z) = p(x_1 \mid z)\, p(x_2 \mid z)$$

Once we know the parents' genes, learning about sibling 1 tells us nothing new about sibling 2 — their genotypes are determined independently from the same source.



However, X_1 and X_2 are *not* unconditionally independent. Before we know the parents' genotype, observing that sibling 1 has a particular trait (say, blue eyes) updates our belief about the parents' genes, which in turn updates our belief about sibling 2. The two children are correlated in the raw data precisely because they share a common cause.

The two scenarios compared.

- **Markov chain** ($X_1 \to X_2 \to X_3$): X_1 and X_3 are unconditionally dependent (the past influences the future) but conditionally independent given X_2 (the present screens off the past). Conditioning *removes* dependence.
- **Shared cause** ($X_1 \leftarrow Z \to X_2$): X_1 and X_2 are unconditionally dependent (correlated through their shared cause) but conditionally independent given Z (knowing the cause makes the effects independent). Conditioning *removes* dependence.

In both cases, conditioning on the right variable blocks the path that creates the dependence. This is the core logic behind Bayes networks, which we develop in the next section.

3.4 Confounding and Causal Effects

The distinction between conditional and unconditional dependence is not merely a mathematical curiosity. It has serious practical consequences when we try to draw conclusions about cause and effect from observational data. The following example illustrates one of the most important pitfalls in applied statistics.

Smoking Habit and Health: A Confounded Comparison

Consider a population of 200 people. Each person is characterized by three variables:

- $Z \in \{\text{young}, \text{old}\}$: age group.
- $X \in \{\text{pipe}, \text{cigarette}\}$: smoking habit.
- $Y \in \{\text{healthy}, \text{unhealthy}\}$: health status.

The population breaks down as follows:

Age Z	Habit X	Total	Unhealthy	$P(Y = \text{u} \mid X, Z)$
Young	Pipe	40	4	0.10
Young	Cigarette	80	8	0.10
Old	Pipe	60	30	0.50
Old	Cigarette	20	10	0.50

Look at the last column. Within each age group, pipe and cigarette smokers have exactly the same probability of being unhealthy: 10% among the young, 50% among the old. **Given age, smoking habit and health are conditionally independent:**

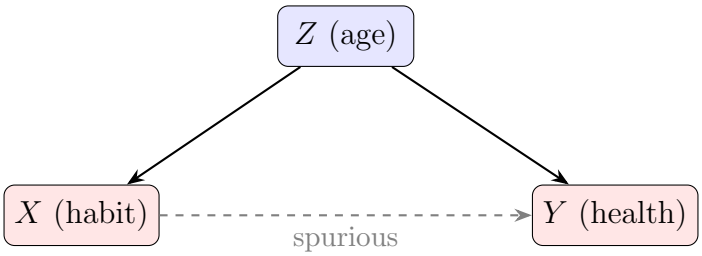
$$P(Y \mid X, Z) = P(Y \mid Z) \qquad \Longleftrightarrow \qquad Y \perp X \mid Z$$

Age Z is the sole driver of health outcomes; within each age group, which type of tobacco you smoke makes no difference.

Now collapse the table and ignore age entirely:

Habit X	Total	Unhealthy	$P(Y = \text{u} \mid X)$
Pipe	100	34	0.34
Cigarette	100	18	0.18

Pipe smokers appear to be almost twice as likely to be unhealthy! A naive analysis would conclude that pipe smoking is more dangerous than cigarette smoking. But we know this is completely wrong: within every age group, the two habits are equally harmful. The spurious association arises because older people — who are unhealthier regardless of smoking habit — happen to prefer pipes in this population (60 out of 80 old people smoke pipes). Age is a **confounder**: a common cause of both the "treatment" (smoking habit) and the "outcome" (health status), as shown in the diagram below.



The dashed arrow from X to Y is not a true causal link — it is a statistical artifact created by the shared cause Z.

Correlation is Not Causation: A Precise Statement

The example above shows that $P(Y \mid X = \text{pipe}) \neq P(Y \mid X = \text{cigarette})$, yet changing a person's smoking habit (holding age fixed) has no effect on their health. How do we define the *causal* effect of X on Y precisely?

The standard framework uses **potential outcomes** (Rubin) or equivalently the **do-calculus** (Pearl). The key idea is to distinguish between *observing* that $X = x$ and *intervening* to set $X = x$.

- $P(Y \mid X = x)$: the probability of Y among people we *observe* to have $X = x$. This is a conditional probability in the usual sense. In our example, this is 0.34 for pipe and 0.18 for cigarette, polluted by confounding.
- $P(Y \mid \text{do}(X = x))$: the probability of Y if we were to *intervene* and force everyone's smoking habit to be x, regardless of their age. This operation, written $\text{do}(X = x)$, severs the arrow from Z to X in the diagram, removing the confounding. The result is a purely causal quantity.

In our example, computing $P(Y \mid \text{do}(X = \text{pipe}))$ requires averaging the health outcome over the age distribution of the *full population*, not just the sub-population of pipe smokers:

$$\begin{aligned} P(Y = \text{u} \mid \text{do}(X = \text{pipe})) &= P(Y = \text{u} \mid X = \text{pipe},\ Z = \text{young})\, P(Z = \text{young}) \\ &\quad + P(Y = \text{u} \mid X = \text{pipe},\ Z = \text{old})\, P(Z = \text{old}) \\ &= 0.10 \times \frac{120}{200} + 0.50 \times \frac{80}{200} = 0.10 \times 0.6 + 0.50 \times 0.4 = 0.26 \end{aligned}$$

By the same calculation, $P(Y = \text{u} \mid \text{do}(X = \text{cigarette})) = 0.26$ as well. The true causal effect is zero: forcing the entire population to switch from pipe to cigarette (or vice versa) would not change the fraction who are unhealthy. The *observed* difference of $0.34 - 0.18 = 0.16$ was entirely due to confounding.

Adjusting for confounders. The formula above, averaging over the confounder Z using the population distribution $P(Z)$, is called the **adjustment formula** or **backdoor adjustment**. It converts the observational conditional $P(Y \mid X, Z)$ into a causal quantity $P(Y \mid \text{do}(X))$ by reweighting over the confounder. This only works when all confounders are measured and included in the adjustment. Unmeasured confounders remain a fundamental challenge in observational studies.

Remark 3.2. Bayesian statistics and causal inference are complementary but distinct. Bayesian inference updates beliefs about unknown quantities using Bayes rule. Causal inference asks what would happen under interventions that change the data-generating process. Neither is a subset of the other. The do-operator and the adjustment formula belong to causal inference; we mention them here because the confounding example is one of the clearest illustrations of why conditional and unconditional dependence can point in opposite directions.

Exercise 3.3. In the population of 200 people above, verify by direct computation that $Y \perp X \mid Z$: show that $P(Y = \text{unhealthy} \mid X, Z)$ is the same for pipe and cigarette smokers within each age group. Then verify that Y and X are *not* marginally independent by computing $P(Y = \text{unhealthy} \mid X = \text{pipe})$ and $P(Y = \text{unhealthy} \mid X = \text{cigarette})$ from the collapsed table.

Exercise 3.4. Suppose the age distribution in the population were reversed: 80 young people and 120 old people, with the same within-group conditional probabilities as above, but now with $P(X = \text{pipe} \mid \text{young}) = 0.6$ and $P(X = \text{pipe} \mid \text{old}) = 0.4$. Reconstruct the full table, compute the marginal $P(Y = \text{u} \mid X)$ for pipe and cigarette smokers, and verify that the adjusted causal effect $P(Y \mid \text{do}(X))$ is still the same for both habits.

3.5 Bayes Networks

Extending Factorization to Many Variables

In Chapter 2 we derived the factorization rule for two variables:

$$p(x, y) = p(x)\, p(y \mid x)$$

This extends to any number of variables by applying the same logic repeatedly. For three variables X, Y, Z, write the joint as a product of a marginal and two conditionals:

$$\begin{aligned} p(x, y, z) &= p(x, y)\, p(z \mid x, y) \\ &= p(x)\, p(y \mid x)\, p(z \mid x, y) \end{aligned}$$

For n variables $X_1, X_2, \ldots, X_n$, applying the same step $n - 1$ times gives the **general chain rule**:

General Chain Rule:

$$p(x_1, \ldots, x_n) = p(x_1)\, p(x_2 \mid x_1)\, p(x_3 \mid x_1, x_2) \cdots p(x_n \mid x_1, \ldots, x_{n-1}) = \prod_{j=1}^{n} p(x_j \mid x_1, \ldots, x_{j-1})$$

This identity holds exactly for any joint distribution and any ordering of the variables.

The chain rule is exact, but it is computationally ruinous. Each conditional $p(x_j \mid x_1, \ldots, x_{j-1})$ depends on all $j-1$ preceding variables. For binary variables, the table for this conditional has 2^{j-1} rows. In a model with 30 binary variables, the last term alone requires a table with $2^{29} \approx 500$ million rows. Storing and computing with the full chain rule is exponential in n — impossible in practice.

The solution is to exploit the structure of real systems. In most problems, each variable has only a small number of *direct* causes. Once those direct causes are known, the variable is independent of everything else. The chain rule can then be dramatically simplified.

Simplifying with Conditional Independence

Recall the two patterns of conditional independence from the previous section.

Pattern 1: Markov chain (immediate cause). Suppose $X_1 \to X_2 \to X_3$: the present X_2 is the immediate cause of the future X_3, and the past X_1 is a remote cause. Once the present is known, the past is irrelevant:

$$X_3 \perp X_1 \mid X_2 \qquad \Rightarrow \qquad p(x_3 \mid x_1, x_2) = p(x_3 \mid x_2)$$

The joint simplifies from the full chain rule to:

$$p(x_1, x_2, x_3) = p(x_1)\, p(x_2 \mid x_1)\, p(x_3 \mid x_1, x_2) = p(x_1)\, p(x_2 \mid x_1)\, p(x_3 \mid x_2)$$

Pattern 2: Shared cause. Suppose $X \leftarrow Z \to Y$: a common cause Z independently generates two effects X and Y. Once the cause is known, the two effects carry no information about each other:

$$X \perp Y \mid Z \qquad \Rightarrow \qquad p(x, y \mid z) = p(x \mid z)\, p(y \mid z)$$

The joint simplifies to:

$$p(x, y, z) = p(z)\, p(x \mid z)\, p(y \mid z)$$

In both patterns, the joint probability is a product of *local* terms: each variable is conditioned only on its **parents** — its direct causes in the causal graph. A **Bayes network** generalizes this idea to any combination of variables and causal structures.

Bayes Network. A Bayes network over variables $X_1, \ldots, X_n$ specifies:

1. A **directed acyclic graph (DAG)** whose nodes are $X_1, \ldots, X_n$. An arrow from X_i to X_j means X_i is a direct cause of X_j. Write $\text{pa}(X_j)$ for the **parents** of X_j.

2. A **conditional probability table (CPT)** for each variable, specifying $p(X_j \mid \text{pa}(X_j))$. Root nodes (with no parents) have unconditional distributions $p(X_j)$.

The joint distribution factorizes as:

$$\boxed{p(x_1, \ldots, x_n) = \prod_{j=1}^{n} p(x_j \mid \text{pa}(x_j))}$$

Each term depends only on the variable's direct causes, not on the entire history of preceding variables.

The savings are dramatic. If each variable has at most k parents, every CPT has at most 2^k rows (for binary variables). The total storage is $O(n \cdot 2^k)$ — linear in n for fixed k — compared to $O(2^n)$ for the full joint table. With 30 binary variables and at most 3 parents each, we need at most $30 \times 8 = 240$ numbers instead of $2^{30} \approx 10^9$.

One Million People Through a Network

The one million people picture extends directly to Bayes networks. Place the million people at the root nodes, distributed according to their priors. Then run each person forward through the network: at each node, independently draw a value from the CPT given the values already assigned to that node's parents. The person passes through every node in topological order (parents before children), and at the end carries a complete configuration $(x_1, \ldots, x_n)$.

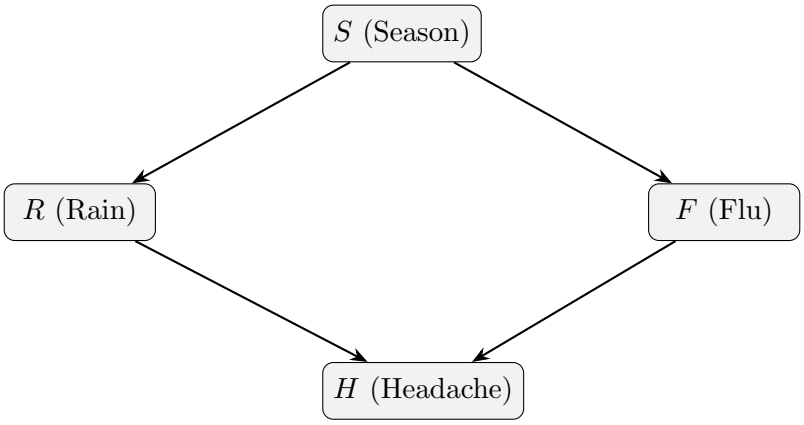
The fraction of people in configuration $(x_1, \ldots, x_n)$ is exactly $\prod_j p(x_j \mid \text{pa}(x_j))$ — the joint probability. The forward physical direction (each cause independently generating its effects) produces the joint distribution automatically, by construction. Inference in the mental direction is then counting: restrict to the people whose observed variables match the evidence, and ask what fraction of those have the query variable in the desired state.

3.6 A Concrete Example: Season, Rain, Flu, and Headache

We now build and use a complete Bayes network with explicit numbers. The network has four binary variables:

- S: Season (0 = winter, 1 = summer)
- R: Rain (0 = no rain, 1 = rain)
- F: Flu (0 = no flu, 1 = flu)
- H: Headache (0 = none, 1 = headache)

The causal structure is shown below. Season is a shared cause of both rain and flu. Both rain and flu contribute independently to headaches.



The joint distribution factorizes along this graph:

$$p(S, R, F, H) = p(S)\, p(R \mid S)\, p(F \mid S)\, p(H \mid R, F)$$

Note carefully what the structure implies. Season is the only common cause of Rain and Flu, so $R \perp F \mid S$: given the season, rain and flu are conditionally independent. Without knowing the season, however, they are correlated — both are more likely in winter. Headache has two direct causes, R and F, and no parents beyond those, so $H \perp S \mid R, F$: once we know whether it rained and whether there is flu, the season gives no additional information about headaches.

The Conditional Probability Tables

Season (no parents):

S	$p(S)$
Winter ($S = 0$)	0.50
Summer ($S = 1$)	0.50

Rain (parent: S):

	$p(R = 0 \mid S)$	$p(R = 1 \mid S)$
Winter ($S = 0$)	0.20	0.80
Summer ($S = 1$)	0.80	0.20

Flu (parent: S):

	$p(F = 0 \mid S)$	$p(F = 1 \mid S)$
Winter ($S = 0$)	0.40	0.60
Summer ($S = 1$)	0.90	0.10

Headache (parents: R, F):

R	F		$p(H = 0 \mid R, F)$	$p(H = 1 \mid R, F)$
0	0	(no rain, no flu)	0.95	0.05
1	0	(rain only)	0.60	0.40
0	1	(flu only)	0.30	0.70
1	1	(rain and flu)	0.10	0.90

Rain alone gives a 40% chance of headache; flu alone gives 70%; both together raises it to 90%; neither leaves only a 5% baseline chance.

One Million People Through This Network

Start with one million people. The season CPT places 500,000 in winter and 500,000 in summer.

Assign Rain. Within winter: $0.8 \times 500 = 400$k get rain, $0.2 \times 500 = 100$k do not. Within summer: $0.2 \times 500 = 100$k get rain, $0.8 \times 500 = 400$k do not. The marginal:

$$p(R = 1) = \frac{400{,}000 + 100{,}000}{1{,}000{,}000} = 0.50$$

Assign Flu. Within winter: $0.6 \times 500 = 300$k get flu. Within summer: $0.1 \times 500 = 50$k get flu. The marginal:

$$p(F = 1) = \frac{300{,}000 + 50{,}000}{1{,}000{,}000} = 0.35$$

Because $R \perp F \mid S$, within each season Rain and Flu are assigned independently. We multiply the within-season fractions directly to get the three-way joint. The table below shows people in thousands ($\times 1000$):

Season	Rain	No flu ($F = 0$)	Flu ($F = 1$)	Row total
Winter	No rain	$500 \times 0.2 \times 0.4 = 40$	$500 \times 0.2 \times 0.6 = 60$	100
Winter	Rain	$500 \times 0.8 \times 0.4 = 160$	$500 \times 0.8 \times 0.6 = 240$	400
Summer	No rain	$500 \times 0.8 \times 0.9 = 360$	$500 \times 0.8 \times 0.1 = 40$	400
Summer	Rain	$500 \times 0.2 \times 0.9 = 90$	$500 \times 0.2 \times 0.1 = 10$	100
Column total		650	350	1,000

Assign Headache. For each of the 8 groups, multiply by the relevant entry in the headache CPT. The column "$H = 1$" gives the number of people (in thousands) who have both the given (Season, Rain, Flu) combination *and* a headache:

S	R	F	People (k)	$p(H = 1 \mid R, F)$	$H = 1$ (k)
Winter	No rain	No flu	40	0.05	2
Winter	No rain	Flu	60	0.70	42
Winter	Rain	No flu	160	0.40	64
Winter	Rain	Flu	240	0.90	216
Summer	No rain	No flu	360	0.05	18
Summer	No rain	Flu	40	0.70	28
Summer	Rain	No flu	90	0.40	36
Summer	Rain	Flu	10	0.90	9
Total			1,000		415

So 415,000 of the million people have a headache: $p(H = 1) = 0.415$.

Inference: What Caused My Headache?

A person wakes up with a headache ($H = 1$). What is the probability they have flu?

In the one million people picture, we restrict attention to the 415,000 people with headaches — the column $H = 1$ in the table above. Among those, how many have flu? The flu rows are:

Winter/No rain/Flu: 42k
Winter/Rain/Flu: 216k
Summer/No rain/Flu: 28k
Summer/Rain/Flu: 9k
Total with flu and headache: 295k

Therefore:

$$p(F = 1 \mid H = 1) = \frac{295{,}000}{415{,}000} = \frac{295}{415} \approx 0.711$$

Before observing any symptoms, the prior probability of flu was $p(F = 1) = 0.35$. A headache more than doubles it to 71.1%. Similarly, the rain rows with headache are $64 + 216 + 36 + 9 = 325$k, giving:

$$p(R = 1 \mid H = 1) = \frac{325}{415} \approx 0.783$$

Both possible causes (rain and flu) become much more probable after observing the headache. This is Bayes rule running in the mental direction: from effect back to causes.

Explaining Away

Now suppose we additionally learn that it is raining ($R = 1$). How does this affect the probability of flu?

Restrict to the 325,000 people with both headache and rain. Among those, the flu rows are Winter/Rain/Flu (216k) and Summer/Rain/Flu (9k), totalling 225,000:

$$p(F = 1 \mid H = 1,\ R = 1) = \frac{225}{325} \approx 0.692$$

Knowing it rained *reduces* the flu probability from 71.1% to 69.2%. Rain has "explained away" part of the headache: some of the evidence that pointed toward flu can now be credited to rain instead.

Conversely, if we learn there is no rain ($R = 0$), the headache is harder to explain, and flu becomes more likely. The 90,000 people with headache and no rain include $42 + 28 = 70$k with flu:

$$p(F = 1 \mid H = 1,\ R = 0) = \frac{70}{90} \approx 0.778$$

Explaining away. When two causes (R and F) share a common effect (H), conditioning on that effect creates a competition between the causes. If one is confirmed, the other becomes less necessary; if one is ruled out, the other becomes more likely. The physical processes generating R and F are completely unrelated — they share only the common cause S, not any direct connection. The dependence between R and F is created entirely by conditioning on their common effect H. This is a fundamental property of causal inference and a key reason why naive intuitions about probability can be misleading.

Toward Efficient Computation: Belief Propagation

The calculation above required tracking 8 groups of people. For a network with n binary variables, the joint table has 2^n rows. For our four-variable network that is

$2^4 = 16$ rows; for a twenty-variable network, over one million. Materializing the full joint table quickly becomes infeasible.

Fortunately, the factored structure of the Bayes network enables dramatically more efficient inference. Instead of working with the full joint, we can pass **messages** between adjacent nodes in the graph. Each message is a small summary — one number per state of the variable — of the information flowing along that edge. At each node, the incoming messages from all neighbors are combined to produce the exact posterior marginal at that node, without ever constructing the full joint. This algorithm is called **belief propagation** or **message passing**.

For tree-structured networks (networks whose undirected skeleton has no cycles), belief propagation is exact and runs in time *linear* in the number of variables — a dramatic improvement over the exponential cost of enumerating all configurations. For networks with cycles, approximate variants (loopy belief propagation) are widely and successfully used in practice, underpinning algorithms from error-correcting codes to natural language processing.

The conceptual foundation in all cases is the same one we have worked through in this section: conditional independence allows computation to be organized locally, one node and one edge at a time, rather than globally over all possible joint configurations.

Exercise 3.5. Using the joint table computed above, find $p(S = 0 \mid H = 1)$: the probability it is winter given that a person has a headache. Among the 415,000 people with headaches, how many are in winter? Does a headache make winter more or less likely compared to the prior $p(S = 0) = 0.5$? Explain intuitively why.

Exercise 3.6. Verify directly from the table that R and F are not marginally independent: compute $p(R = 1, F = 1)$ from the three-way table and check whether it equals $p(R = 1) \times p(F = 1)$. Then verify conditional independence given season: check that $p(R = 1, F = 1 \mid S = 0) = p(R = 1 \mid S = 0) \times p(F = 1 \mid S = 0)$.

Exercise 3.7. Suppose a person has a headache and we are told they have flu ($F = 1$). Compute $p(S = 0 \mid H = 1, F = 1)$ — the probability it is winter given headache and flu. Does knowing they have flu make winter more or less likely compared to $p(S = 0 \mid H = 1)$ from the first exercise? Explain the direction of the change in terms of the causal structure of the network.

Chapter 4

From Bayes Rule to Bayesian Statistics

Bayes rule is a theorem of pure mathematics. It follows from the definition of conditional probability and requires no assumptions beyond the axioms of probability theory. Everyone — frequentist, Bayesian, and agnostic alike — accepts it without reservation. Yet one of the most consequential and contested moves in the history of statistics is the application of Bayes rule not to observable random events, but to *unknown parameters*: quantities like the probability that a coin lands heads, the speed of light, or the efficacy of a drug. This chapter traces exactly where that move occurs, why it is philosophically nontrivial, and why it is also, in many situations, the most sensible thing to do.

We begin with the history.

4.1 Thomas Bayes and the Posthumous Paper

Thomas Bayes was born around 1701 in London, the son of a Nonconformist minister. He followed his father into the ministry, serving the Presbyterian congregation at Mount Sion chapel in Tunbridge Wells, Kent, for most of his adult life. He was also a Fellow of the Royal Society, elected in 1742, though he published almost nothing during his lifetime. He died in April 1761, leaving behind an unpublished manuscript.

The manuscript was found among Bayes' papers by his friend Richard Price, a Welsh moral philosopher and Fellow of the Royal Society in his own right. Price recognized its significance immediately. He edited the essay, added an introduction of his own, and submitted it to the Royal Society. It was read before the Society in 1763 and published that same year in the *Philosophical Transactions of the Royal Society* under the title *An Essay towards Solving a Problem in the Doctrine of Chances*. The author was listed as "Mr. Bayes, F.R.S.," two years after his death.

Price's introduction is remarkable for its candor about the motivation. He writes that the paper addresses a problem that had long seemed intractable: given that an

event has occurred a certain number of times and failed a certain number of times, what can be said about the probability of its occurring in future trials? This is not a question about known probabilities but about *unknown* ones. It is the problem of inference from data.

The specific problem Bayes posed, and solved, was this. A billiard table has length 1. A ball is thrown uniformly at random and lands at some position $p \in [0, 1]$. This position p is unknown. Subsequently, n more balls are thrown, each independently landing to the left of the first ball with probability p. The observer counts how many of the n balls land to the left, obtaining a number k, and from this count must reason about the unknown p. The question is: what can the observer infer about p given k?

What makes this problem special is that the unknown p is itself the outcome of a physical randomization — the throw of the first ball. Bayes and Price were careful about this. The prior distribution over p is not a statement of subjective belief pulled from thin air; it is the physical distribution from which p was actually drawn. The prior $p \sim \text{Uniform}[0, 1]$ is legitimate because it corresponds to a real, repeatable physical act.

Remark 4.1. It is worth pausing on the intellectual climate of 1763. The problem Bayes solved — how to learn about an unknown probability from data — is so fundamental that it is hard to imagine it ever being open. Yet the mathematical tools for addressing it did not exist before Bayes. The only known results involved computing probabilities forward from a known model. Bayes' insight was to run the calculation backward: from the observed data to the unknown model parameter. The paper went largely unnoticed for several decades, until Pierre-Simon Laplace independently rediscovered and vastly extended the same ideas.

4.2 Laplace and the Birth Statistics of Paris

Pierre-Simon Laplace (1749–1827) was one of the dominant mathematical scientists of his era: a central figure in celestial mechanics, probability theory, and mathematical physics. He rediscovered Bayesian reasoning independently and applied it with a breadth and ambition that dwarfed Bayes' original paper.

In his 1812 masterwork *Théorie analytique des probabilités*, Laplace applied Bayesian inference to a question about the birth process in Paris: is the probability that a birth results in a boy equal to the probability that it results in a girl, or is one sex genuinely more likely? He had access to the birth records of Paris spanning several decades. For the period he analyzed, 251,527 boys and 241,945 girls had been born, a total of 493,472 births. The observed fraction of boys was $251{,}527/493{,}472 \approx 0.5097$.

The question sounds simple: just compute the fraction. But Laplace asked something deeper: what is the probability that the true underlying probability of a male birth exceeds 1/2? And what is our uncertainty about this probability given the data?

To answer this, Laplace placed a probability distribution over the unknown parameter p — the true probability that any birth results in a boy. He used a uniform

prior $p \sim \text{Uniform}[0,1]$ and, after observing $k = 251{,}527$ boys in $n = 493{,}472$ births, computed the posterior distribution of p.

The crucial philosophical difference from Bayes. In Bayes' billiard ball problem, the unknown p was physically generated by throwing the first ball. A frequentist is comfortable with this: the prior has a real frequency interpretation. In Laplace's problem, no one threw a ball to generate p. The parameter p is a propensity of the birth process — a fixed feature of nature. There is no physical randomization that produces p, no urn from which it was drawn. The prior Uniform$[0,1]$ represents the observer's uncertainty about p before seeing the data, not any physical frequency.

This is the **philosophical leap** at the heart of Bayesian statistics: applying probability to quantities that are not outcomes of repeatable experiments. Laplace made this leap explicitly and unapologetically. He argued that probability quantifies *knowledge*, not just physical randomness. An observer uncertain about a fixed but unknown quantity is in exactly the same epistemic position as an observer uncertain about the outcome of a random experiment; probability theory should apply equally to both.

Laplace computed the posterior probability that $p > 1/2$ given the birth data to be approximately $1 - 10^{-42}$: essentially certain. The data overwhelmingly support a genuine male excess in births. This is a statement about the observer's updated knowledge, grounded in Bayes rule, but requiring the prior to make it precise.

We now develop the mathematics that underlies both Bayes' and Laplace's calculations.

4.3 Mathematical Preparation: Gamma and Beta Functions

Before working through the billiard ball posterior, we need three mathematical tools: the Gamma function, the Beta function, and the Beta distribution. These appear ubiquitously in Bayesian calculations with count data.

The Gamma Function

The factorial $n! = n \times (n-1) \times \cdots \times 2 \times 1$ is defined for non-negative integers. Many formulas in probability require extending factorial to non-integer arguments. The **Gamma function** does this.

Definition 4.2 (Gamma Function)**.** For $\alpha > 0$, the Gamma function is:

$$\Gamma(\alpha) = \int_0^\infty x^{\alpha-1} e^{-x}\, dx$$

This integral converges for all $\alpha > 0$. Let us establish its key properties.

Property 1: Recursion. For any $\alpha > 0$:

$$\Gamma(\alpha + 1) = \alpha\, \Gamma(\alpha)$$

Proof. Integrate by parts with $u = x^\alpha$ and $dv = e^{-x}dx$:

$$\Gamma(\alpha+1) = \int_0^\infty x^\alpha e^{-x}\,dx = \left[-x^\alpha e^{-x}\right]_0^\infty + \int_0^\infty \alpha x^{\alpha-1}e^{-x}\,dx = 0 + \alpha\,\Gamma(\alpha)$$

The boundary term vanishes because e^{-x} decays faster than any power of x grows. □

Property 2: Integers. For any positive integer n:

$$\Gamma(n) = (n-1)!$$

Proof. We verify the base case: $\Gamma(1) = \int_0^\infty e^{-x}\,dx = 1 = 0!$. Then apply the recursion n times: $\Gamma(n) = (n-1)\,\Gamma(n-1) = (n-1)(n-2)\,\Gamma(n-2) = \cdots = (n-1)!$. □

The Gamma function therefore extends factorial to all positive reals. Some important values:

$$\Gamma(1) = 1, \quad \Gamma(2) = 1, \quad \Gamma(3) = 2, \quad \Gamma(4) = 6, \quad \Gamma(n) = (n-1)!$$
$$\Gamma\left(\tfrac{1}{2}\right) = \sqrt{\pi}$$

The value $\Gamma(1/2) = \sqrt{\pi}$ is proved by the Gaussian integral and will reappear when we discuss the normal distribution.

The Beta Function

Definition 4.3 (Beta Function)**.** For $\alpha, \beta > 0$, the Beta function is:

$$B(\alpha,\beta) = \int_0^1 x^{\alpha-1}(1-x)^{\beta-1}\,dx$$

The Beta function is the normalizing constant that will appear in the Beta distribution. Its key property connects it to the Gamma function.

Proposition 4.4.

$$B(\alpha,\beta) = \frac{\Gamma(\alpha)\,\Gamma(\beta)}{\Gamma(\alpha+\beta)}$$

Proof sketch. Write $\Gamma(\alpha)\,\Gamma(\beta)$ as a double integral, substitute $x = u/(u+v)$ and $s = u+v$, and recognize the resulting integrals as $B(\alpha,\beta)$ and $\Gamma(\alpha+\beta)$. The algebra is straightforward but lengthy; the key step is the change of variables that separates the two integrals. □

For integer arguments this gives a clean formula. Let $\alpha = k+1$ and $\beta = n-k+1$ for non-negative integers $k \le n$:

$$B(k+1, n-k+1) = \frac{\Gamma(k+1)\,\Gamma(n-k+1)}{\Gamma(n+2)} = \frac{k!\,(n-k)!}{(n+1)!} = \frac{1}{(n+1)\binom{n}{k}}$$

This identity is the key to computing the normalizing constant in Bayes' calculation.

The Beta Distribution

Definition 4.5 (Beta Distribution)**.** A random variable P has the **Beta distribution** with parameters $\alpha > 0$ and $\beta > 0$, written $P \sim \text{Beta}(\alpha, \beta)$, if its density is:

$$p(x) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1}(1-x)^{\beta-1}, \qquad x \in [0, 1]$$

The factor $1/B(\alpha, \beta)$ ensures the density integrates to one over $[0, 1]$. In practice we almost always write:

$$p(x) \propto x^{\alpha-1}(1-x)^{\beta-1}$$

and handle the normalizing constant separately.

Mean and variance. Direct integration using the Beta function identity gives:

$$\mathbb{E}[P] = \frac{\alpha}{\alpha+\beta}, \qquad \text{Var}(P) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$$

Special cases.

- Beta$(1, 1)$: the density is $p(x) \propto 1$, which is the Uniform$[0, 1]$ distribution. Both parameters equal to 1 means no prior information favoring any value of x.
- Beta(α, β) with large α and β: the distribution concentrates near its mean $\alpha/(\alpha+\beta)$. Large parameters represent strong prior knowledge.
- Beta$(1/2, 1/2)$: the Jeffreys prior for a binomial proportion (discussed later), which places more mass near 0 and 1 than near $1/2$.

The Beta distribution is the natural family for probabilities — unknown quantities constrained to $[0, 1]$ — and it will be the posterior distribution in every binomial problem we encounter.

4.4 Bayes' Billiard Ball: The Posterior is Beta

We now solve Bayes' problem. The setup: a billiard table of length 1. A servant throws the first ball uniformly at random; it lands at position $p \sim \text{Uniform}[0, 1]$. Then n more balls are thrown independently, each landing to the left of the first with probability p. The servant reports only the count k: the number of balls that landed to the left. The question: given k, what is the posterior distribution of p?

The Likelihood

Given p, the count k follows a Binomial distribution: each of the n balls independently lands left with probability p, so:

$$p(k \mid p) = \binom{n}{k} p^k (1-p)^{n-k}$$

This is the **physical direction**: the known parameter p generates the observed count k.

The Prior

Since the first ball was thrown uniformly, $p \sim \text{Uniform}[0,1] = \text{Beta}(1,1)$:

$$p(p) = 1, \qquad p \in [0,1]$$

Deriving the Posterior

Apply Bayes rule:

$$p(p \mid k) = \frac{p(p)\, p(k \mid p)}{p(k)}$$

The numerator is:

$$p(p)\, p(k \mid p) = 1 \cdot \binom{n}{k} p^k (1-p)^{n-k} \propto p^k (1-p)^{n-k}$$

The denominator $p(k)$ does not depend on p; it is a normalizing constant. We compute it using the Beta function:

$$p(k) = \int_0^1 \binom{n}{k} p^k (1-p)^{n-k}\, dp = \binom{n}{k} B(k+1, n-k+1) = \binom{n}{k} \cdot \frac{k!\,(n-k)!}{(n+1)!} = \frac{1}{n+1}$$

Therefore the posterior density is:

$$p(p \mid k) = \frac{p^k (1-p)^{n-k}}{B(k+1,\, n-k+1)}$$

The posterior is Beta:

$$p \mid k,\, n \;\sim\; \text{Beta}(k+1,\; n-k+1)$$

The prior $\text{Uniform}[0,1] = \text{Beta}(1,1)$ gets updated to $\text{Beta}(k+1, n-k+1)$ after observing k successes in n trials. The parameters of the Beta distribution keep a running tally: $k+1$ counts the left-landings plus the one unit of prior, and $(n-k)+1$ counts the right-landings plus the one unit of prior.

Posterior mean and Laplace's rule of succession.

$$\mathbb{E}[p \mid k, n] = \frac{k+1}{n+2}$$

This is never exactly 0 or 1. Even after observing n consecutive successes ($k = n$), the posterior mean is $(n+1)/(n+2) < 1$. No finite amount of data makes you

completely certain. Laplace called this the **rule of succession**: the probability of a success on the next trial, given k successes in n past trials, is $(k+1)/(n+2)$. It is a theorem, derived from Bayes rule and the uniform prior. It implies, for example, that the sun having risen every day for n days suggests but does not guarantee it will rise tomorrow.

Approximate Bayesian Computation: The Tireless Servant

There is a beautiful operational way to understand the posterior that requires no calculus at all.

Approximate Bayesian Computation (ABC):

1. Throw the first ball to sample $p \sim \text{Uniform}[0,1]$.
2. Throw n more balls and count how many land left. Call the count $\tilde{k}$.
3. **If** $\tilde{k} = k_{\text{observed}}$, **keep** this p. Otherwise discard it and return to Step 1.
4. The distribution of the kept p values is exactly the posterior $p(p \mid k_{\text{observed}})$.

This is simply the one million people picture applied to a parameter. Among all people who send k_{observed} balls left, what is the distribution of their starting positions p? That is the posterior. The servant is doing nothing more sophisticated than conditional counting. The servant must be tireless because most throws produce $\tilde{k} \neq k_{\text{observed}}$, so most samples are discarded; but each kept sample is an exact draw from the posterior.

Is the Prior Legitimate?

In Bayes' billiard problem, the prior $p \sim \text{Uniform}[0,1]$ is physically meaningful: p is the actual landing position of a real ball, generated by a real physical act. A committed frequentist can accept this prior because it corresponds to a repeatable experiment. The prior has a frequency interpretation.

This legitimacy is specific to Bayes' setup. The moment we apply the same mathematics to a parameter that is not generated by a physical randomization — such as Laplace's birth probability — we have gone further. The mathematical calculation is identical; the philosophical justification is different. We return to this question after developing the necessary examples.

4.5 The Beta-Binomial Model: A Complete Analysis

We now work through a concrete numerical example to make the Beta-Binomial posterior fully tangible.

Setup. Suppose $n = 10$ balls are thrown after the first. We observe $k = 7$ landing to the left.

Prior. $p \sim \text{Beta}(1, 1) = \text{Uniform}[0, 1]$. Before seeing any data, every position for the first ball is equally likely.

Posterior. $p \mid k = 7, n = 10 \sim \text{Beta}(8, 4)$.

The posterior mean is $8/12 = 2/3$. The posterior is concentrated around $p \approx 2/3$, reflecting the fact that 7 out of 10 balls landed left. But it retains substantial spread because $n = 10$ is small. The table below shows how the posterior changes as more data arrives:

n	k	Posterior	Post. mean	Post. std
0	0	Beta$(1, 1)$	0.500	0.289
5	3	Beta$(4, 3)$	0.571	0.181
10	7	Beta$(8, 4)$	0.667	0.132
20	14	Beta$(15, 7)$	0.682	0.096
50	35	Beta$(36, 16)$	0.692	0.061
100	70	Beta$(71, 31)$	0.696	0.044

Three features stand out. First, the posterior mean moves toward the true value as data accumulates — this is consistency. Second, the posterior standard deviation shrinks as $1/\sqrt{n}$ — this is the statistical learning rate. Third, the prior mean of 0.5 becomes completely irrelevant for large n: the data overwhelms any reasonable prior. This will be a general theme.

Conjugacy. The reason the prior Beta$(1, 1)$ produces a posterior Beta$(k + 1, n - k + 1)$ is that the Beta family is **conjugate** to the Binomial likelihood: a Beta prior combined with Binomial data always yields a Beta posterior. The prior and posterior are in the same parametric family; only the parameters change. Conjugacy makes computation exact and gives the update rule a natural interpretation: the prior parameters (α, β) act like pseudo-counts of prior successes and failures, and the data simply adds real counts on top:

$$\underbrace{\text{Beta}(\alpha, \beta)}_{\text{prior}} + \text{data } (k, n) \longrightarrow \underbrace{\text{Beta}(\alpha + k,\ \beta + n - k)}_{\text{posterior}}$$

4.6 Laplace's Gender Problem: The Philosophical Leap

We now return to Laplace's birth data. The model is identical to Bayes', with p now meaning the probability that any birth results in a boy. Laplace observed $k = 251{,}527$ boy births and $m = 241{,}945$ girl births, a total of $n = k + m = 493{,}472$ births.

With the prior $p \sim \text{Uniform}[0, 1] = \text{Beta}(1, 1)$, the posterior is:

$$p \mid k,\, m \;\sim\; \text{Beta}(k + 1,\; m + 1) = \text{Beta}(251{,}528,\; 241{,}946)$$

The posterior mean is:

$$\mathbb{E}[p \mid k, m] = \frac{k+1}{n+2} = \frac{251{,}528}{493{,}474} \approx 0.5097$$

The posterior is concentrated very tightly around 0.5097. The posterior standard deviation is:

$$\text{SD}(p \mid k, m) = \sqrt{\frac{(k+1)(m+1)}{(n+2)^2(n+3)}} \approx \frac{1}{2\sqrt{n}} \approx \frac{1}{2\sqrt{493{,}472}} \approx 0.000712$$

The posterior is a spike. The probability that $p > 1/2$ is:

$$P(p > 1/2 \mid k, m) = \int_{1/2}^{1} \frac{p^k (1-p)^m}{B(k+1, m+1)}\, dp \approx 1 - 10^{-42}$$

Laplace interpreted this as overwhelming evidence that boys are genuinely more likely than girls at birth. More precisely: given the data and the uniform prior, an observer should be nearly certain that the underlying birth probability favors boys.

The philosophical issue. In Bayes' setup, the prior was physically justified. In Laplace's problem, it is not. The parameter p is a fixed property of human biology. There is no urn from which p was drawn. The prior Uniform$[0, 1]$ is not a frequency statement — it is a statement about Laplace's knowledge before seeing the data. Laplace was explicit about this: he interpreted probability as a measure of *reasonable belief*, not physical frequency.

This is the moment at which Bayesian statistics diverges from Bayes' original, frequency-compatible formulation. The mathematics is identical. The object to which it is applied has changed fundamentally. This philosophical step is necessary, unavoidable, and — Laplace argued — the only coherent way to answer the question "what do the data tell us about p?"

4.7 The Speed of Light: When Repetition is Impossible

We now move to a setting where the philosophical tension is even sharper: estimating a physical constant. The physicist Albert Michelson made careful measurements of

the speed of light in the 1880s. Let θ denote the true speed of light (a fixed physical constant: there is one universe, one value of θ). Each measurement y_i is subject to random error:

$$y_i = \theta + \varepsilon_i, \qquad \varepsilon_i \sim \mathcal{N}(0, \sigma^2), \quad i = 1, \ldots, n$$

Here σ^2 is the measurement error variance, which we treat as known for now. The data $y_1, \ldots, y_n$ are n independent noisy measurements of θ.

A frequentist treats θ as a fixed unknown and builds a confidence interval for the estimation procedure. A Bayesian places a prior distribution over θ, updates it with the data, and reports the posterior. The frequentist approach produces a procedure with good frequency properties; the Bayesian approach produces a probability statement about θ directly. For this, a prior is required.

The Gaussian Distribution

Before deriving the posterior, we need the Gaussian distribution carefully.

Definition 4.6 (Gaussian Distribution)**.** A random variable X follows the **Gaussian (normal) distribution** with mean μ and variance σ^2, written $X \sim \mathcal{N}(\mu, \sigma^2)$, if its density is:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \qquad x \in \mathbb{R}$$

The factor $1/\sqrt{2\pi\sigma^2}$ ensures the density integrates to one; this follows from the Gaussian integral $\int_{-\infty}^{\infty} e^{-x^2/2}\,dx = \sqrt{2\pi}$. The distribution is symmetric around μ, has standard deviation σ, and falls off exponentially fast in the tails.

The log density. Taking the logarithm:

$$\log p(x) = -\frac{(x-\mu)^2}{2\sigma^2} - \frac{1}{2}\log(2\pi\sigma^2)$$

The log density is a downward-opening parabola in x with maximum at $x = \mu$. This quadratic structure is what makes Gaussian calculations tractable: sums of log-Gaussian densities remain quadratic, and quadratics can be combined by completing the square.

Key fact: identifying a Gaussian from its log density. Suppose we know that $\log f(x) = ax - \frac{b}{2}x^2 + C$ as a function of x, for constants a, $b > 0$, and C. Then f is the density of a Gaussian with:

$$\text{mean} = \frac{a}{b}, \qquad \text{variance} = \frac{1}{b}$$

This is the key computational tool for all Gaussian posterior calculations. We identify the linear and quadratic coefficients in the log density, read off the mean and variance, and the normalizing constant takes care of itself.

The Prior

We place a Gaussian prior on θ:

$$\theta \sim \mathcal{N}(\mu_0, \tau^2)$$

Here μ_0 is our best guess for θ before seeing the data (perhaps from prior physical theory), and τ^2 quantifies how uncertain we are about that guess. The precision of the prior is $1/\tau^2$: a large precision (small τ^2) means a strong, tight prior; a small precision (large τ^2) means a weak, diffuse prior.

The Likelihood

The n measurements are independent Gaussians:

$$p(y_1, \ldots, y_n \mid \theta) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - \theta)^2}{2\sigma^2}\right)$$

Taking the log:

$$\begin{aligned} \log p(y_1, \ldots, y_n \mid \theta) &= -\sum_{i=1}^{n} \frac{(y_i - \theta)^2}{2\sigma^2} + C_1 \\ &= -\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \theta)^2 + C_1 \end{aligned}$$

where C_1 absorbs terms that do not depend on θ. Expand the square:

$$\sum_{i=1}^{n} (y_i - \theta)^2 = \sum_{i=1}^{n} y_i^2 - 2\theta \sum_{i=1}^{n} y_i + n\theta^2 = n\theta^2 - 2n\bar{y}\theta + \sum y_i^2$$

where $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ is the sample mean. Therefore:

$$\log p(y_1, \ldots, y_n \mid \theta) = -\frac{n}{2\sigma^2}(\theta^2 - 2\bar{y}\theta) + C_2 = \frac{n\bar{y}}{\sigma^2}\theta - \frac{n}{2\sigma^2}\theta^2 + C_2$$

The sufficient statistic is $\bar{y}$: the entire dataset enters the log-likelihood only through the sample mean.

Deriving the Posterior Step by Step

By Bayes rule:

$$\log p(\theta \mid y_1, \ldots, y_n) = \log p(\theta) + \log p(y_1, \ldots, y_n \mid \theta) + C_3$$

Step 1: Write out the log prior.

$$\log p(\theta) = -\frac{(\theta - \mu_0)^2}{2\tau^2} + C_4 = -\frac{\theta^2 - 2\mu_0\theta}{2\tau^2} + C_5 = \frac{\mu_0}{\tau^2}\theta - \frac{1}{2\tau^2}\theta^2 + C_5$$

Step 2: Add log prior and log likelihood.

$$\begin{aligned}\log p(\theta \mid \mathbf{y}) &= \left(\frac{\mu_0}{\tau^2}\theta - \frac{1}{2\tau^2}\theta^2\right) + \left(\frac{n\bar{y}}{\sigma^2}\theta - \frac{n}{2\sigma^2}\theta^2\right) + C \\ &= \underbrace{\left(\frac{\mu_0}{\tau^2} + \frac{n\bar{y}}{\sigma^2}\right)}_{a}\theta \; - \; \frac{1}{2}\underbrace{\left(\frac{1}{\tau^2} + \frac{n}{\sigma^2}\right)}_{b}\theta^2 + C\end{aligned}$$

Step 3: Read off the Gaussian parameters. The log posterior is $a\theta - \frac{b}{2}\theta^2 + C$, which is Gaussian with:

$$\text{precision} = b = \frac{1}{\tau^2} + \frac{n}{\sigma^2}, \qquad \text{mean} = \frac{a}{b} = \frac{\dfrac{\mu_0}{\tau^2} + \dfrac{n\bar{y}}{\sigma^2}}{\dfrac{1}{\tau^2} + \dfrac{n}{\sigma^2}}$$

Define $\tau_n^2 = 1/b$ (posterior variance) and $\mu_n = a/b$ (posterior mean). The result:

Gaussian-Gaussian Posterior:

$$\theta \mid y_1, \ldots, y_n \;\sim\; \mathcal{N}(\mu_n,\, \tau_n^2)$$

where:

$$\frac{1}{\tau_n^2} = \frac{1}{\tau^2} + \frac{n}{\sigma^2}, \qquad \mu_n = \frac{\dfrac{1}{\tau^2}\mu_0 + \dfrac{n}{\sigma^2}\bar{y}}{\dfrac{1}{\tau^2} + \dfrac{n}{\sigma^2}}$$

The **posterior precision** is the sum of the prior precision and the data precision. The **posterior mean** is the precision-weighted average of the prior mean and the sample mean.

Interpreting the Result

Precision is additive. The posterior precision $1/\tau_n^2 = 1/\tau^2 + n/\sigma^2$ is the prior precision plus the data precision. Every new measurement adds $1/\sigma^2$ to the total precision. Information from independent sources adds. This is intuitive: more data, more precision; noisier data (σ^2 large), less precision per measurement.

The posterior mean as a weighted average. Write $w_0 = (1/\tau^2)/(1/\tau^2 + n/\sigma^2)$ for the weight on the prior. Then $\mu_n = w_0\mu_0 + (1 - w_0)\bar{y}$: a convex combination of prior mean and data mean, with weights proportional to their respective precisions. The more precise source receives more weight.

Limiting cases. Consider what happens as n grows:

- $n = 0$: $\mu_n = \mu_0$, $\tau_n^2 = \tau^2$. No data, posterior equals prior.
- $n \to \infty$: $\mu_n \to \bar{y}$, $\tau_n^2 \to 0$. Data overwhelms the prior; the posterior collapses onto $\bar{y}$.
- $\tau^2 \to \infty$ (diffuse prior): $w_0 \to 0$, $\mu_n \to \bar{y}$. When the prior is very uncertain, even small amounts of data dominate.
- $\sigma^2 \to \infty$ (very noisy measurements): $w_0 \to 1$, $\mu_n \to \mu_0$. When measurements are very noisy, the prior is more informative than the data.

Numerical example. Suppose previous physical theory suggests $\theta = 299{,}800$ km/s with uncertainty $\tau = 100$ km/s. Michelson makes $n = 5$ measurements with noise $\sigma = 50$ km/s. The sample mean is $\bar{y} = 299{,}850$ km/s.

Quantity	Formula	Value
Prior precision	$1/\tau^2$	$1/10{,}000$
Data precision	n/σ^2	$5/2{,}500 = 1/500$
Post. precision	$1/10{,}000 + 1/500$	$21/10{,}000$
τ_n^2	$10{,}000/21$	476 km^2/s^2
τ_n	$\sqrt{476}$	21.8 km/s
μ_n	$(1/10{,}000 \times 299{,}800 + 1/500 \times 299{,}850)/(21/10{,}000)$	299,848 km/s

The posterior mean of 299,848 km/s is close to the sample mean (299,850) because the data precision 1/500 is twenty times larger than the prior precision 1/10,000. The prior has little influence here. The posterior standard deviation is 21.8 km/s, tighter than either the prior (100 km/s) or the per-measurement noise (50 km/s).

The credible interval. A 95% **posterior credible interval** for θ is:

$$[\mu_n - 1.96\tau_n,\ \mu_n + 1.96\tau_n] = [299{,}848 \pm 42.7] = [299{,}805,\ 299{,}891] \text{ km/s}$$

This interval contains θ with posterior probability 95%. This is what practitioners intuitively *want* an interval to mean: "the true value lies in here with 95% probability." A frequentist confidence interval does not mean this. It means: "the procedure that produced this interval covers the true θ in 95% of hypothetical repetitions." For a fixed constant like the speed of light, "hypothetical repetitions" is a conceptually strained notion. The Bayesian credible interval is the more natural object.

When Repetition is Impossible

The speed of light example sharply exposes the limits of the frequentist framework. There is one speed of light. We cannot repeat the universe to check whether our

confidence interval procedure would cover θ in 95% of alternate universes. The frequentist confidence statement is about the procedure, not about θ; it does not assign a probability to the event $\theta \in [a, b]$ for any specific a, b.

A Bayesian is comfortable making a direct probability statement about θ, because Bayesian probability quantifies the observer's knowledge, not the frequency of events in repeated experiments. The prior encodes what was known before the measurements; the posterior encodes what is known after. Whether θ "has a distribution" in any physical sense is irrelevant — what matters is that the observer is uncertain about θ, and probability theory is the correct calculus for reasoning under uncertainty.

This is a genuine philosophical difference. It is not resolved by data, because it concerns the *interpretation* of probability statements, not their numerical content. For many practical purposes, the two approaches give similar answers with enough data. But for unique events — the speed of light, the probability that a specific patient survives a specific treatment, the fairness of a specific election — the Bayesian framework is the only one that makes direct probability statements about the quantity of interest.

4.8 Reinterpreting Repetition: The ABC Perspective

The previous section raised a genuine difficulty. In Laplace's problem, the parameter p is a fixed property of the birth process. In Michelson's experiment, θ is a fixed physical constant. Neither was generated by a physical randomization. So what does it mean to say $p \sim \text{Uniform}[0, 1]$ or $\theta \sim \mathcal{N}(\mu_0, \tau^2)$? If the prior is not a frequency, what is it?

The ABC procedure offers a clarifying answer. Let us revisit it carefully from this angle.

Solving an Equation by Trying

Consider a simple analogy. Suppose you want to solve the equation:

$$f(x) = y_{\text{observed}}$$

for an unknown x, given an observed value y_{observed}. You do not know how to invert f analytically. One strategy is to *try* different values of x, evaluate $f(x)$, and keep the values of x for which $f(x)$ is close to y_{observed}. The set of x values that pass this test is your solution set — your best answer for what x could be.

Before you start trying, you need to decide *how* to try. Which values of x do you try first? If you know from other considerations that x is likely to be near 5, you try more values near 5 and fewer near 1000. If you truly know nothing, you try values spread evenly across the plausible range. The distribution over which you try values is exactly the prior.

Bayesian inference is this procedure applied to statistical models. The model $p(y \mid \theta)$ is the function f: it maps a parameter value θ to a distribution over observable data y. The observed data y_{observed} is the target. The question is: which values of θ are consistent with what was observed?

ABC as Organized Trying

The ABC procedure makes this precise:

ABC as organized trying:

1. **Try** a value of θ by drawing it from the prior: $\theta^{(s)} \sim p(\theta)$.
2. **Simulate** data from the model using this value: $\tilde{y}^{(s)} \sim p(y \mid \theta^{(s)})$.
3. **Keep** $\theta^{(s)}$ if the simulated data matches the observed data: $\tilde{y}^{(s)} = y_{\text{observed}}$. Otherwise discard it.
4. **Repeat** steps 1–3 many times. The collection of kept values $\theta^{(s)}$ is a sample from the posterior $p(\theta \mid y_{\text{observed}})$.

The prior $p(\theta)$ governs *how we try*: it determines the distribution over which candidate values $\theta^{(s)}$ are drawn. It is not a statement about the frequency with which nature generates θ. It is a statement about the observer's strategy for exploring the space of possible solutions.

What the Prior Means, Precisely

The prior quantifies the observer's uncertainty about θ *before seeing any data*. This uncertainty is real and operational: it determines which values of θ the observer considers plausible and therefore worth trying. Two observers with different prior knowledge will try different regions of the parameter space more intensively, and will reach different posteriors from the same data. This is not a flaw — it is an honest reflection of the fact that two observers with genuinely different prior information should reach different conclusions.

The prior is *not*:

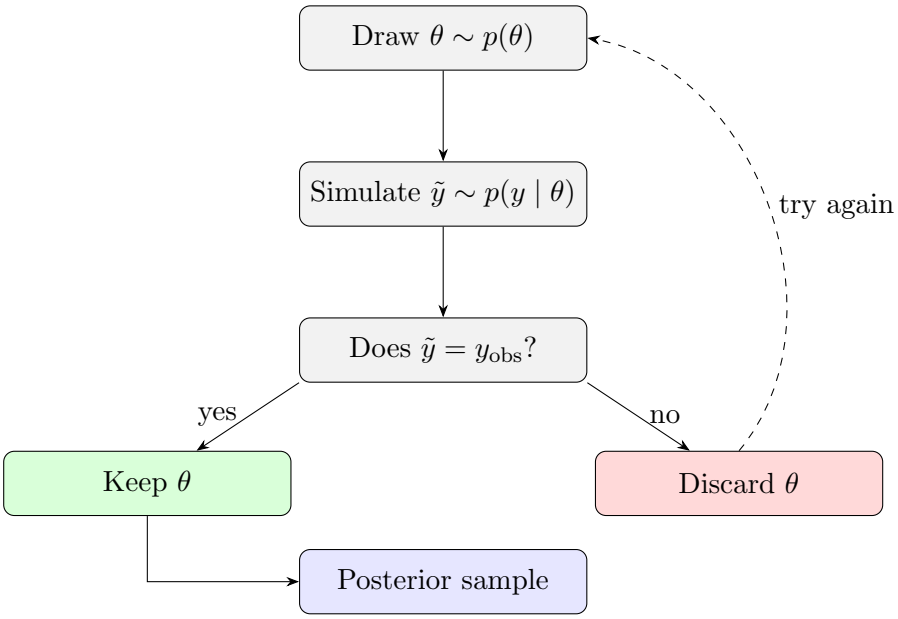
- A claim that θ was generated by a physical randomization.
- A statement about the long-run frequency with which nature produces various values of θ.
- A subjective whim that can be chosen arbitrarily.

The prior *is*:

- A complete description of the observer's uncertainty about θ before data arrives.
- A strategy for exploring the parameter space: which values of θ to try more, and which to try less.
- Constrained by coherence: if the observer has genuine prior knowledge (physical theory, previous experiments, domain expertise), the prior should reflect it.

The Posterior as a Filtered Distribution

The ABC procedure clarifies what the posterior is as well. Start with one million candidate values of θ, each drawn from the prior. Run each through the model to generate a simulated dataset. Keep only those whose simulated dataset matches the observed one. The distribution of the surviving candidates is the posterior.



The posterior is exactly the conditional distribution of θ given that the model, when run with θ, produces the observed data. This is Bayes rule — nothing more. The ABC procedure is a computational method for sampling from this conditional distribution by direct simulation.

Returning to Laplace and Michelson

In Laplace's birth problem, the prior Uniform$[0, 1]$ says: before seeing the data, I have no reason to prefer any value of p in $[0, 1]$ over any other. In ABC terms: I try candidate values of p uniformly across $[0, 1]$. After filtering to those values that produce a simulated dataset resembling 251,527 boys in 493,472 births, the surviving

candidates are tightly clustered near $p \approx 0.5097$. The posterior is the distribution of these survivors. No claim about the physical origin of p is required.

In Michelson's experiment, the prior $\mathcal{N}(\mu_0, \tau^2)$ says: based on existing physical theory and previous experiments, I expect the speed of light to be near μ_0, with uncertainty τ. In ABC terms: I try candidate values of θ more densely near μ_0 and more sparsely far from it. After filtering to those candidates that produce simulated measurements resembling Michelson's actual readings, the posterior tells me exactly what I should believe about θ given everything I know. The prior is an encoding of genuine prior knowledge, not a frequency.

The reinterpretation. The prior does not require θ to be the outcome of a random experiment. It requires only that the observer is uncertain about θ and can describe that uncertainty as a probability distribution over candidate values. The posterior is then the result of filtering those candidates through the data: keeping those consistent with what was observed, discarding the rest. This is the ABC procedure, and it is mathematically identical to Bayes rule. The prior is not a frequency. It is the observer's strategy for organized exploration of the unknown.

Remark 4.7. The ABC procedure also reveals why the prior matters less as data accumulates. With many observations, the filter $\tilde{y} = y_{\text{obs}}$ is very tight: only candidate values of θ in a narrow range near the true value survive. Whether the prior put a lot of weight on that region or a little becomes irrelevant — as long as the prior was not zero there, the posterior will concentrate in the same place regardless. This is posterior consistency: with enough data, all reasonable priors lead to the same posterior. The observer's prior strategy is eventually overwhelmed by the evidence.

Exercise 4.8. Implement the ABC procedure for the Beta-Binomial model with $n = 10$ and $k = 7$:

1. Draw $p^{(s)} \sim \text{Uniform}[0, 1]$ for $s = 1, \ldots, 100{,}000$.
2. For each $p^{(s)}$, simulate $\tilde{k}^{(s)} \sim \text{Binomial}(10, p^{(s)})$.
3. Keep $p^{(s)}$ if $\tilde{k}^{(s)} = 7$.
4. Plot a histogram of the kept values and overlay the $\text{Beta}(8, 4)$ density. They should match closely.
5. Approximately what fraction of the 100,000 trials are kept? Explain why this fraction is approximately $1/(n + 1) = 1/11$.

The ABC Argument Extends Naturally to Prediction

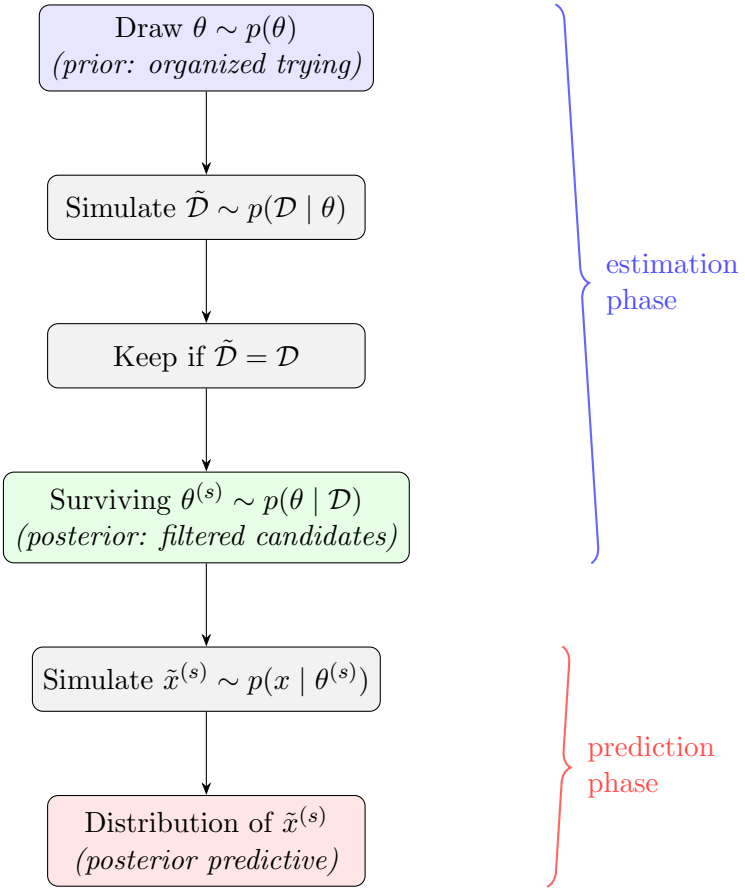
The ABC picture does not stop at parameter estimation. It extends seamlessly to prediction, and in doing so it reveals the full coherence of the Bayesian framework in a way no classical argument achieves.

Recall the ABC loop. We start with one million candidate values θ drawn from the prior $p(\theta)$. We filter them by the data $\mathcal{D}$: keep only those for which the simulated data matches what was observed. The surviving candidates follow the posterior $p(\theta \mid \mathcal{D})$. So far this is estimation.

Now suppose a new observation x_{new} arrives. What should we predict? The surviving candidates — those consistent with $\mathcal{D}$ — are now our pool of plausible parameter values. For each survivor $\theta^{(s)}$, simulate a new observation $\tilde{x}^{(s)} \sim p(x \mid \theta^{(s)})$ from the model. The distribution of these simulated future observations is the **posterior predictive distribution**:

$$p(x_{\text{new}} \mid \mathcal{D}) = \int p(x_{\text{new}} \mid \theta)\, p(\theta \mid \mathcal{D})\, d\theta \approx \frac{1}{S} \sum_{s=1}^{S} p(x_{\text{new}} \mid \theta^{(s)})$$

The logic is the same as before: among all (θ, x_{new}) pairs generated by the model, keep only those whose θ was consistent with the past data. The marginal distribution of the retained x_{new} values is the posterior predictive. No new machinery is required. The posterior is simply the prior for the next question.



Now suppose more data $\mathcal{D}'$ arrives after the prediction. We filter the surviving candidates again: keep only those $\theta^{(s)}$ for which a new simulation from the model also

matches $\mathcal{D}'$. The result is $p(\theta \mid \mathcal{D}, \mathcal{D}')$ — the posterior updated on all data seen so far. The posterior from the first round becomes the prior for the second round, exactly as Bayes rule requires.

Sequential learning in the ABC picture.

1. Start with the prior $p(\theta)$: the initial search strategy, encoding what is known before any data.
2. Observe $\mathcal{D}$. Filter candidates. The posterior $p(\theta \mid \mathcal{D})$ is the new pool of plausible values.
3. Predict future observations by simulating forward from the pool. The posterior predictive $p(x_{\text{new}} \mid \mathcal{D})$ is the distribution of plausible futures.
4. Observe $\mathcal{D}'$. Filter again. The posterior $p(\theta \mid \mathcal{D}, \mathcal{D}')$ is the further refined pool.
5. Repeat. Each posterior is the prior for the next question.

At every stage, the same operation — filter candidates by data, simulate forward for predictions — is applied. The prior, the posterior, and the posterior predictive are not three separate concepts requiring three separate justifications. They are three moments in a single coherent cycle of learning.

This sequential coherence is something the classical arguments cannot easily provide. Cox's theorem establishes that probability is the right language for uncertainty; it does not explain why today's posterior should be tomorrow's prior. Dutch book arguments establish coherence at a single moment; they do not naturally generate a learning rule. Savage's axioms apply to a single decision problem; extending them to sequential decisions requires additional structure.

The ABC picture makes sequential coherence obvious. Of course today's posterior is tomorrow's prior: it is simply the updated pool of candidate solutions, ready to be filtered again by new evidence. The Bayesian learning cycle is not a philosophical commitment. It is the natural consequence of organized search. You start with what you know. You filter by what you see. You predict by simulating forward from what survived. You filter again when more arrives. Each step is the same operation, applied to an ever-more-refined pool of candidates. Nothing more is required.

This also clarifies the role of the prior in prediction. The prior's influence on predictions diminishes as data accumulates, because the pool of surviving candidates concentrates. An analyst worried that their prior choice will distort predictions can take comfort: with enough data, any reasonable prior leads to the same pool of survivors and therefore the same predictions. The prior matters most for the first prediction, before much data has been seen — which is precisely when prior knowledge is most valuable and when having an explicit, thoughtful search strategy is most important.

4.9 Uninformative Priors and Jeffreys Prior

A persistent concern about Bayesian statistics is the role of the prior. If two analysts use different priors, they will reach different posteriors. Is Bayesian inference therefore subjective? And is there a canonical prior that represents "knowing nothing"?

The naive answer: uniform prior. The most obvious choice for a prior representing ignorance is the uniform distribution: every value of θ equally likely. But uniform on what scale? A uniform prior on θ is not uniform on θ^2, or on $\log\theta$. The choice of parameterization should not secretly determine the prior.

Example 4.9 (The reparametrization problem)**.** Suppose $\theta \sim \text{Uniform}[0,1]$. Define $\phi = -\log\theta$ (so θ is a probability and ϕ is the corresponding mean waiting time in a geometric model). Then by the change-of-variables formula, ϕ has density:

$$p(\phi) = p(\theta)\left|\frac{d\theta}{d\phi}\right| = 1 \cdot e^{-\phi} = e^{-\phi}, \quad \phi > 0$$

This is an Exponential(1) distribution, not uniform. The "uninformative" uniform prior on θ is highly informative about ϕ: it says the mean waiting time is almost certainly small. The same state of ignorance, expressed in two different parameterizations, produces two different priors. Uniform is not truly uninformative.

Jeffreys prior. Harold Jeffreys proposed a resolution: define the prior to be *invariant under reparametrization*. This requires the prior to scale with the *geometry* of the model as measured by the Fisher information:

$$p_J(\theta) \propto \sqrt{I(\theta)}, \qquad I(\theta) = -\mathbb{E}_y\left[\frac{\partial^2}{\partial\theta^2}\log p(y \mid \theta)\right]$$

The Fisher information $I(\theta)$ measures how quickly the likelihood changes with θ — how much information a single observation carries about θ. If a small change in θ produces a large change in the likelihood, observations are very informative about θ and the Jeffreys prior puts less mass there. The Jeffreys prior is flatter where the likelihood is curved, and more concentrated where the likelihood is flat.

Under a reparametrization $\phi = g(\theta)$, the Fisher information transforms as $I_\phi(\phi) = I_\theta(\theta)/(g'(\theta))^2$. The Jeffreys prior transforms as a density should, so $p_J(\phi) \propto \sqrt{I_\phi(\phi)}$ holds in both parameterizations simultaneously. It is the unique prior (up to scaling) with this property.

Example 4.10 (Jeffreys priors for common models)**.**

- **Gaussian mean, σ^2 known.** $I(\theta) = 1/\sigma^2$ (constant), so $p_J(\theta) \propto 1$: the uniform prior is the Jeffreys prior. Uniform is only genuinely uninformative for location parameters.
- **Gaussian variance, μ known.** $I(\sigma^2) = 1/(2\sigma^4)$, so $p_J(\sigma^2) \propto 1/\sigma^2$. A uniform prior on σ^2 would be informative in disguise; the Jeffreys prior says the scale of uncertainty is uniform on the log scale.

- **Binomial probability** p**.** $I(p) = 1/(p(1-p))$, so $p_J(p) \propto p^{-1/2}(1-p)^{-1/2}$: the Beta$(1/2, 1/2)$ distribution. This places more mass near $p = 0$ and $p = 1$ than the uniform Beta$(1, 1)$, reflecting the fact that observations near $p = 0$ or $p = 1$ are very informative about p.

Uninformative does not mean uniform. Ignorance has a geometry, determined by the information structure of the model. The Jeffreys prior is the prior that respects this geometry — it contains the same information about θ regardless of how θ is parameterized. For location parameters (like a Gaussian mean), it happens to be uniform. For scale parameters (like a variance), it is not.

Exercise 4.11. Let $y_1, \ldots, y_5 = 299{,}810;\ 299{,}850;\ 299{,}890;\ 299{,}840;\ 299{,}860$ (in km/s). Assume $\sigma = 50$ km/s.

1. Compute the sample mean $\bar{y}$.

2. Using the prior $\theta \sim \mathcal{N}(299{,}800,\ 100^2)$, compute the posterior mean μ_n and posterior standard deviation τ_n.

3. How much does the posterior mean change if instead you use a more diffuse prior $\theta \sim \mathcal{N}(299{,}800,\ 1000^2)$? Explain why the change is small in terms of precisions.

4. Construct a 95% posterior credible interval for θ under both priors.

Exercise 4.12. Let $P \sim \text{Beta}(\alpha, \beta)$.

1. Verify that $\mathbb{E}[P] = \alpha/(\alpha+\beta)$ by computing $\int_0^1 x \cdot x^{\alpha-1}(1-x)^{\beta-1}\,dx/B(\alpha,\beta)$ using the Beta function identity.

2. Show that as $\alpha + \beta \to \infty$ with $\alpha/(\alpha+\beta) \to p_0$ fixed, the Beta distribution concentrates around p_0. (Hint: compute the variance and show it goes to zero.)

3. For the birth data with $k = 251{,}527$ and $m = 241{,}945$, compute the posterior mean, posterior standard deviation, and the approximate probability that $p > 0.51$ under the uniform prior. (Hint: use the normal approximation to the Beta for large parameters.)

Exercise 4.13. Show that the Jeffreys prior for the Binomial proportion is Beta$(1/2, 1/2)$ by computing the Fisher information $I(p) = -\mathbb{E}_y[\partial^2 \log p(y \mid p)/\partial p^2]$ for $y \sim \text{Binomial}(n, p)$ and verifying that $\sqrt{I(p)} \propto p^{-1/2}(1-p)^{-1/2}$.

4.10 Credible Intervals, Confidence Intervals, and Calibration

The speed of light example produces a posterior distribution $\theta \mid y_1, \ldots, y_n \sim \mathcal{N}(\mu_n, \tau_n^2)$. From this posterior we can immediately read off an interval that contains θ with 95% posterior probability:

$$[\mu_n - 1.96\tau_n,\ \mu_n + 1.96\tau_n]$$

This is called a **95% credible interval**. Its meaning is direct: given the data and the prior, the probability that the true speed of light lies in this interval is 95%. This is what every scientist instinctively wants an interval to mean. It is also, precisely, what a frequentist confidence interval does *not* mean. The difference between the two is not a technicality. It is a conceptual gulf that goes to the heart of what probability means.

What a Confidence Interval Actually Means

A **95% frequentist confidence interval** $[L(y), U(y)]$ is a procedure, not a statement about the unknown parameter. It is constructed so that, if the experiment were repeated infinitely many times, 95% of the resulting intervals would contain the true θ. Formally:

$$P_\theta(L(y) \leq \theta \leq U(y)) = 0.95 \quad \text{for all } \theta$$

The probability here is over the random variable y: over hypothetical repetitions of the experiment. For any specific observed interval $[L(y_{\text{obs}}), U(y_{\text{obs}})]$, the true θ is either inside it or not. There is no randomness left. The frequentist framework does not assign a probability to this specific interval containing θ.

The standard Gaussian confidence interval for the speed of light, after observing n measurements with sample mean $\bar{y}$ and known noise σ^2, is:

$$\left[\bar{y} - 1.96\frac{\sigma}{\sqrt{n}},\ \bar{y} + 1.96\frac{\sigma}{\sqrt{n}}\right]$$

This interval is wider when n is small (less data, more uncertainty) and narrower when n is large (more data, less uncertainty). It has the correct frequentist coverage: in 95% of imagined repetitions of Michelson's experiment, the interval would contain the true speed of light. But it says nothing about whether *this specific interval*, computed from *this specific set of measurements*, contains the true value. The probability statement is about the procedure, not the result.

The Conceptual Problem with Confidence Intervals

The confidence interval interpretation requires imagining repetitions of the experiment. For many problems, this is a natural thought experiment. For the speed of light, it is strained.

The speed of light is a fixed physical constant. There is one universe, one value of θ. Michelson ran his experiment a finite number of times, in specific conditions, in a specific year. What does it mean to imagine repeating this experiment "infinitely many times"? Does each repetition use the same equipment? The same atmospheric conditions? A different physicist? The frequentist confidence statement is about a sequence of hypothetical experiments that were never run and never will be.

More practically: a scientist who computes the confidence interval [299,805, 299,891] km/s and is asked "does the true speed of light lie in this interval?" wants to answer "yes, with 95% probability." The frequentist framework forbids this answer. The interval either contains θ or it does not; we just do not know which. The 95% refers to the long-run behavior of the procedure, not to this specific interval.

This is not a philosophical quibble. It is a practical problem. Scientists, doctors, engineers, and policymakers who use statistical intervals almost universally intend them as probability statements about the unknown quantity. The frequentist framework cannot deliver this. The Bayesian framework can.

Credible Intervals Say What You Think They Say

The Bayesian credible interval has exactly the interpretation that practitioners want. The statement:

$$P(\theta \in [\mu_n - 1.96\tau_n,\ \mu_n + 1.96\tau_n] \mid y_1, \ldots, y_n) = 0.95$$

means: given the data I have observed and my prior knowledge, there is a 95% probability that the true speed of light lies in this interval. This is a statement about θ, conditional on the data. It is the answer to the question the scientist is actually asking.

The price is the prior. The credible interval depends on the prior $\mathcal{N}(\mu_0, \tau^2)$ that was placed on θ. A different prior produces a different credible interval. This is not a weakness — it is an honest acknowledgment that the interval reflects both the data and the prior knowledge. When the prior is diffuse (τ^2 large), the credible interval and the confidence interval are numerically very close: the data dominates and the prior's influence on the interval is small. When the prior is informative, the credible interval can be substantially different — and the Bayesian would argue, more accurate, because it uses more information.

Credible interval vs. confidence interval.

Credible interval	Confidence interval
$P(\theta \in [a, b] \mid \mathcal{D}) = 0.95$	$P(L(y) \leq \theta \leq U(y)) = 0.95$
Probability is over θ, given data	Probability is over y, for all θ
Statement about the unknown parameter	Statement about the procedure
Requires a prior	Does not require a prior
Says what practitioners think it says	Does not say what practitioners think it says
Numerically similar to CI when prior is diffuse	Numerically similar to credible interval when prior is diffuse

A Thought Experiment: The Specific Interval

Here is a simple thought experiment that sharpens the distinction. Suppose we compute the confidence interval and find:

$$[299{,}805,\ 299{,}891] \text{ km/s}$$

A student asks: "Is the true speed of light in this interval?" The frequentist must answer: "I cannot say. The interval either contains the true value or it does not. What I can say is that my procedure produces intervals that contain the true value 95% of the time." The Bayesian answers: "Yes, with 95% posterior probability, given the data and my prior."

Now suppose the student asks a follow-up: "I have just been told by a reliable source that the true speed of light is 300,000 km/s, which is outside your interval. Should I update my belief that the interval contains the true value?" The frequentist framework has nothing to say: the coverage probability is a property of the procedure, not of this specific interval, and it does not update on new information. The Bayesian framework updates immediately: the posterior probability that the true value lies in the interval drops sharply. Probability, in the Bayesian framework, is a state of knowledge that updates with evidence. This is what probability should do.

Calibration

The concept of **calibration** connects the Bayesian and frequentist perspectives on intervals. A procedure for producing q-credible intervals is **calibrated** if, over many

problems and datasets, the fraction of intervals that actually contain the true value is q.

Formally, let θ_j be the true parameter in problem j, and let $[a_j(y), b_j(y)]$ be the computed q-interval for that problem. The procedure is calibrated if:

$$\frac{1}{J}\sum_{j=1}^{J} \mathbf{1}[\theta_j \in [a_j, b_j]] \to q \quad \text{as } J \to \infty$$

Are Bayesian credible intervals calibrated? Yes, under the right conditions. If the prior $p(\theta)$ correctly describes the distribution of true parameter values across problems — that is, if θ_j are genuinely drawn from $p(\theta)$ — then Bayesian credible intervals are exactly calibrated. This follows from the tower property of conditional expectation:

$$P(\theta \in [a(y), b(y)]) = \mathbb{E}_\theta[P(\theta \in [a(y), b(y)] \mid \theta)] = \mathbb{E}_\theta[q] = q$$

The outer expectation is over the distribution of θ across problems; the inner probability is the posterior coverage, which is q by construction.

When the prior is wrong. If the prior does not match the true distribution of θ values, the credible intervals may be miscalibrated. A prior that is too narrow produces credible intervals that are too short: the true value falls outside more often than $1-q$. A prior that is too wide produces intervals that are too long: overcoverage. Calibration is a diagnostic for prior misspecification.

Frequentist confidence intervals are always calibrated (by construction), but at the cost of not being probability statements about θ. Bayesian credible intervals are probability statements about θ, and they are calibrated when the prior is correct. The two frameworks achieve calibration by different mechanisms and make different guarantees.

Calibration in one sentence. A procedure is calibrated if its stated confidence matches its actual long-run coverage. Bayesian credible intervals are calibrated when the prior matches reality; frequentist confidence intervals are calibrated by construction but make no probability statement about the specific interval in hand.

Practical Implications

The distinction matters most in three situations.

Unique events. For a fixed physical constant, a specific patient's treatment outcome, or the result of a single election, there is no natural repetition. The frequentist confidence interval's coverage guarantee is a statement about a hypothetical sequence of experiments that does not exist. The Bayesian credible interval is the only framework that can make a direct probability statement about the outcome of interest.

Decision making. When an interval is used to make a decision — approve a drug, launch a product, set a policy — the decision maker needs to know the probability that

the parameter exceeds a threshold, given the data. This is a posterior probability. A confidence interval does not provide it. A credible interval does.

Communicating uncertainty. The credible interval $[a, b]$ means "I believe the true value is in $[a, b]$ with probability q," which is what every non-statistician thinks a statistical interval means. Teaching people to use confidence intervals correctly requires explaining a subtle and counterintuitive distinction that most users never fully internalize. The credible interval communicates uncertainty in the way that humans naturally think about uncertainty.

Exercise 4.14. Suppose $y_1, \ldots, y_5$ are measurements of a physical constant θ with known noise $\sigma = 2$. The sample mean is $\bar{y} = 10.3$.

1. Compute the 95% frequentist confidence interval for θ.
2. Using the prior $\theta \sim \mathcal{N}(10, 4)$, compute the 95% Bayesian credible interval.
3. Are the two intervals the same? Which is wider, and why?
4. Write one sentence that correctly interprets the confidence interval, and one sentence that correctly interprets the credible interval. Which sentence would a non-statistician find more natural?

Exercise 4.15. A medical test has sensitivity 0.95 (probability of a positive test given disease) and specificity 0.90 (probability of a negative test given no disease). The disease prevalence is 0.01.

1. Compute $P(\text{disease} \mid \text{positive test})$ using Bayes rule.
2. A doctor reports: "The test has 95% sensitivity, so a positive result means the patient almost certainly has the disease." What error is the doctor making? How does this relate to the difference between $P(\text{test} \mid \text{disease})$ and $P(\text{disease} \mid \text{test})$?
3. The doctor's error is an example of confusing the physical direction with the mental direction. Explain this in the language of Chapter 2.
4. A 95% confidence interval is sometimes misinterpreted as "a 95% probability that the true value is in the interval." Explain why this is the same type of error as the doctor's: confusing a statement about the data given the parameter with a statement about the parameter given the data.

4.11 Inverse Probability: From Laplace to Fisher

The word "Bayesian" was not used by Bayes, by Laplace, or by any of the great astronomers and mathematicians who developed statistical inference in the eighteenth and nineteenth centuries. They used a different term: **inverse probability**. The

terminology is revealing. Direct probability asks: given a known mechanism, what outcomes are likely? Inverse probability asks: given observed outcomes, what can we infer about the unknown mechanism? The direction is reversed. This is precisely the mental direction of Chapter 2 — reasoning from effect back to cause — and it was the central preoccupation of the greatest quantitative scientists of the era.

The story of inverse probability is the story of how Bayesian reasoning became the dominant framework for scientific inference, then was dismantled, and then slowly rebuilt. It runs from Laplace's celestial mechanics to Gauss's least squares to Fisher's likelihood, and it is one of the most consequential intellectual trajectories in the history of science.

Laplace and the Weight of Evidence

Laplace was not modest about the scope of inverse probability. In his *Théorie analytique des probabilités* (1812) and the popular *Essai philosophique sur les probabilités* (1814), he presented probability theory as the mathematical formalization of common sense: the tool by which a rational mind updates its beliefs in light of evidence. He applied it everywhere.

The problems Laplace cared about were not toy examples. He wanted to know the mass of Saturn, the shape of the Earth, the probability that the sun would rise tomorrow. These are inverse problems in the most literal sense: the mass of Saturn is not observable, but its gravitational influence on the orbits of its moons is. Given the observed orbital data, what can be inferred about the mass?

Laplace's approach was always the same. Let θ be the unknown quantity (the mass of Saturn, the ellipticity of the Earth, the birth probability of a boy). Let y be the observed data (the orbital periods, the geodetic measurements, the birth records). Place a prior $p(\theta)$ — typically uniform, reflecting ignorance before the data — and compute the posterior:

$$p(\theta \mid y) \propto p(y \mid \theta)\, p(\theta)$$

From the posterior, compute the quantities of interest: the posterior mean as a point estimate, the posterior probability that θ exceeds a threshold, the predictive distribution for future observations. This is the complete programme of Bayesian statistics, fully articulated two centuries ago.

Laplace's calculations were heroic. Without computers, without matrices, often without even the normal distribution in its modern form, he computed posteriors by hand for astronomical datasets with hundreds of observations. His results were accurate. His estimates of the mass of Saturn, the mass of Jupiter, and the ellipticity of the Earth were consistent with modern values. Inverse probability worked.

The central limit theorem as a tool. Laplace recognized that for large datasets, the posterior $p(\theta \mid y)$ concentrates near a single value and becomes approximately Gaussian, regardless of the prior and the likelihood (provided the likelihood is smooth and the data is informative). This is the **Bernstein-von Mises theorem** in its earliest

form. It gave Laplace a practical tool: for large n, compute the mode of the posterior (which is approximately the MLE) and the curvature of the log posterior at the mode (which gives the posterior variance). The posterior is then approximately:

$$p(\theta \mid y) \approx \mathcal{N}\left(\hat{\theta},\ \left[-\frac{\partial^2 \log p(\theta \mid y)}{\partial \theta^2}\bigg|_{\hat{\theta}}\right]^{-1}\right)$$

This is the Laplace approximation, still widely used today in variational inference and approximate Bayesian computation.

Gauss and the Derivation of Least Squares

Carl Friedrich Gauss (1777–1855) arrived at the method of least squares independently of Legendre (who published it first in 1805), and the derivation he gave in his 1809 work *Theoria Motus Corporum Coelestium* — Theory of the Motion of Celestial Bodies — is a masterpiece of inverse probability reasoning.

Gauss's problem was astronomical. Given a set of observations $y_1, \ldots, y_n$ of some celestial quantity (an orbital element, the position of a planet), each subject to random measurement error ε_i, how should the true value θ be estimated?

Gauss's argument proceeded in three steps.

Step 1: The error distribution. Gauss argued that the measurement errors $\varepsilon_i = y_i - \theta$ should follow some distribution $f(\varepsilon)$ that is symmetric, unimodal, and concentrated near zero. He then asked: what distribution f has the property that the most probable value of θ given the observations — that is, the *posterior mode* under a uniform prior — is the arithmetic mean $\bar{y} = \frac{1}{n}\sum y_i$? He regarded the arithmetic mean as the natural, intuitively obvious estimate, and wanted to find the error distribution that justified it.

Step 2: Inverse probability. Under a uniform prior on θ, the posterior is proportional to the likelihood:

$$p(\theta \mid y_1, \ldots, y_n) \propto \prod_{i=1}^{n} f(y_i - \theta)$$

The posterior mode — the most probable value of θ given the data — maximizes this product, or equivalently maximizes:

$$\sum_{i=1}^{n} \log f(y_i - \theta)$$

For the mode to equal $\bar{y}$ for all datasets, differentiating with respect to θ and setting to zero gives:

$$\sum_{i=1}^{n} \frac{f'(y_i - \theta)}{f(y_i - \theta)} = 0 \quad \text{at } \theta = \bar{y}$$

Step 3: The Gaussian emerges. Gauss showed that the unique solution to this functional equation is:

$$f(\varepsilon) \propto \exp\left(-\frac{\varepsilon^2}{2\sigma^2}\right)$$

the Gaussian (normal) distribution. This is the *derivation* of the normal distribution, not its postulation. The normal distribution is the unique error distribution for which the arithmetic mean is the posterior mode under a uniform prior.

Gauss's argument in one sentence. The normal distribution is the error distribution that makes the arithmetic mean the inverse probability estimate. Least squares is the method that finds this estimate. The justification is Bayesian: the arithmetic mean is the posterior mode under a Gaussian likelihood and a uniform prior.

The method of least squares follows immediately: given Gaussian errors, the posterior mode is

$$\hat{\theta} = \arg\max_{\theta} \sum_{i=1}^{n} \log f(y_i - \theta) = \arg\min_{\theta} \sum_{i=1}^{n} (y_i - \theta)^2$$

Minimizing the sum of squared errors is maximizing the posterior under a Gaussian likelihood and a uniform prior. This is MAP estimation, with the uniform prior making MAP equal to MLE.

Gauss generalized. In later work, Gauss extended this to the multivariate case: estimating a vector of orbital parameters β from observations $y = X\beta + \varepsilon$. The posterior mode under a uniform prior on β and Gaussian errors is:

$$\hat{\beta} = (X^\top X)^{-1} X^\top y$$

the ordinary least squares estimator. Gauss derived this formula, proved that it minimized the sum of squared residuals, and showed that it was the best linear unbiased estimator — the result now called the Gauss-Markov theorem. He did all of this using the language and logic of inverse probability.

The Nineteenth Century Consensus

Through the first half of the nineteenth century, inverse probability was not a controversial doctrine. It was the standard framework for quantitative inference in astronomy, geodesy, and physics. The greatest scientists of the era — Laplace, Gauss, Poisson, Quetelet — used it without apology. The prior was almost always taken to be uniform, so in practice the posterior mode coincided with the MLE and the posterior was approximately Gaussian by the Laplace approximation. The Bayesian and frequentist answers were numerically almost identical, and the philosophical question of what the prior meant was not urgently pressed.

The situation changed gradually in the second half of the century, as statisticians began to examine the foundations more carefully. The uniform prior, taken for granted by Laplace and Gauss, began to look less innocent. It was not invariant under reparametrization (as Jeffreys would later formalize). For a parameter on an unbounded space it was improper — it did not integrate to one. And for some problems, different "natural" parameterizations led to different uniform priors and hence different answers. The solidity of inverse probability began to crack.

Fisher and the Demolition of the Prior

Ronald Aylmer Fisher (1890–1962) is the most influential statistician of the twentieth century, and the most consequential opponent of inverse probability. He did not merely criticize the Bayesian approach; he set out to replace it entirely with a new framework that he believed was both more rigorous and more honest.

Fisher's objection was not to Bayes theorem itself — he accepted that as mathematics — but to the prior. Specifically, he argued that in most scientific problems, there is no legitimate prior probability for the unknown parameter θ. The speed of light, the effect of a drug, the coefficient in a regression: these are fixed but unknown constants. They are not outcomes of random experiments. There is no urn from which they are drawn, no physical process that generates them with frequencies that could justify a probability distribution. Assigning a uniform prior, as Laplace and Gauss had done, was not a statement of ignorance; it was, Fisher argued, an arbitrary choice masquerading as a neutral one.

Fisher's alternative was the **likelihood function**. Given data y and model $p(y \mid \theta)$, define:

$$L(\theta) = p(y \mid \theta)$$

viewed as a function of θ for fixed y. The likelihood is not a probability distribution over θ — it does not integrate to one over θ, and Fisher was emphatic that it should not be interpreted as one. It is a function that captures how well each value of θ explains the observed data, without any prior attached.

Maximum likelihood estimation. Fisher proposed the **maximum likelihood estimate** (MLE) as the natural summary of the likelihood:

$$\hat{\theta}_{\text{MLE}} = \arg\max_{\theta} \, L(\theta) = \arg\max_{\theta} \, p(y \mid \theta)$$

This is the value of θ that makes the observed data most probable. Fisher showed that the MLE has remarkable asymptotic properties: it is consistent (converges to the true θ as $n \to \infty$), asymptotically normal (the distribution of $\hat{\theta}_{\text{MLE}}$ is approximately Gaussian for large n), and asymptotically efficient (it achieves the Cramér-Rao lower bound on variance).

Notice that these are exactly the properties Laplace derived for the posterior mode under a uniform prior. This is not a coincidence: for a uniform prior, the MLE and

the MAP estimate are identical. Fisher's MLE is Bayesian inference with the prior stripped away and the remaining machinery reinterpreted as not requiring one.

Fisher information. Fisher also defined the quantity that now bears his name:

$$I(\theta) = \mathbb{E}_y\left[\left(\frac{\partial \log p(y \mid \theta)}{\partial \theta}\right)^2\right] = -\mathbb{E}_y\left[\frac{\partial^2 \log p(y \mid \theta)}{\partial \theta^2}\right]$$

The Fisher information measures the curvature of the log-likelihood at θ: how sharply the likelihood peaks, how much information the data carries about θ. The Cramér-Rao bound states that any unbiased estimator has variance at least $1/(nI(\theta))$. The MLE achieves this bound asymptotically.

Fisher information also reappears in the Jeffreys prior $p_J(\theta) \propto \sqrt{I(\theta)}$: the prior that is invariant under reparametrization. This is the one place where Fisher's concept enters Bayesian statistics directly — and it is the prior that addresses the very problem Fisher identified with uniform priors.

The fiducial distribution. Fisher's own attempt to recover probability statements about parameters without priors was the **fiducial distribution**, introduced in 1930. If the data y is related to θ by $y = g(\theta, \varepsilon)$ for some known function g and random variable ε with known distribution, Fisher proposed "inverting" this relationship to obtain a distribution for θ given y. He called this the fiducial distribution and believed it captured the evidential content of the data without requiring a prior.

The fiducial idea never achieved a satisfactory mathematical foundation. For simple models it coincided with the Bayesian posterior under Jeffreys prior, which undermined Fisher's claim that it was prior-free. For more complex models, it led to contradictions and paradoxes. Fisher defended it passionately until his death, but it was never accepted by the statistical community, and it gradually faded. The episode illustrates how difficult it is to recover the content of Bayesian inference while rejecting its premises.

What Was Lost and What Was Kept

Fisher's revolution was enormously productive. The likelihood function, maximum likelihood estimation, Fisher information, the Cramér-Rao bound, sufficiency, ancillarity, and the analysis of variance are all Fisher's contributions, and they constitute a large fraction of the foundation of modern statistics. The frequentist framework — confidence intervals, hypothesis tests, p-values — was built on Fisher's insights by Neyman and Pearson in the 1930s and became the dominant framework for applied statistics throughout the twentieth century.

But something was also lost. The likelihood function captures how well each θ explains the data, but it does not answer the question the scientist is actually asking: given the data, what should I believe about θ? The MLE gives the single most likely value, but not a probability distribution over values. The confidence interval covers θ in 95% of hypothetical repetitions, but does not give the probability that the specific

interval contains the true value. The framework is precise about the procedure and silent about the parameter.

What was lost was inverse probability: the ability to make probability statements about unknown quantities directly. Laplace could say "the probability that the speed of light is between 299,800 and 299,900 km/s, given these measurements, is 95%." Fisher could not. The price of Fisher's rigor about the prior was the surrender of the most natural thing a scientist wants to say.

The arc from Laplace to Fisher.

	Inverse probability (Laplace, Gauss)	**Likelihood (Fisher)**
Unknown θ	Random variable with a distribution	Fixed but unknown constant
Prior	Uniform (or other)	Rejected as arbitrary
Inference	Posterior $p(\theta \mid y)$	Likelihood $L(\theta) = p(y \mid \theta)$
Point estimate	Posterior mode (= MLE for uniform prior)	MLE
Interval	Posterior credible interval	Confidence interval (procedure)
Probability statement about θ	Yes: $P(\theta \in [a, b] \mid y) = q$	No: coverage is over hypothetical y
Justification for Gaussians	Inverse probability derivation (Gauss 1809)	Asymptotic theory

The Return

Fisher's framework dominated statistics from roughly 1925 to 1980. The Bayesian approach survived, advocated by a small number of statisticians — Harold Jeffreys, Leonard Savage, Dennis Lindley — but remained a minority position, viewed by the mainstream as philosophically suspect and computationally intractable.

Two developments changed this. First, the development of MCMC methods in the 1980s and 1990s made Bayesian computation feasible for complex models. Second, and more subtly, the rise of machine learning created a vast new class of problems — model selection, prediction, uncertainty quantification, sequential decision-making — for which the frequentist framework gave awkward or incomplete answers. The Bayesian framework gave natural ones.

Today, the two frameworks coexist. Most applied statisticians use frequentist methods for inference and hypothesis testing, and Bayesian methods for prediction, model

selection, and problems with genuine prior information. Most machine learning practitioners use the likelihood for training (MLE, or equivalently cross-entropy minimization) and the posterior predictive for generation and uncertainty quantification. The distinction between Bayes theorem (mathematics) and Bayesian philosophy (contested) that this book emphasizes reflects the actual practice of the field: everyone uses Bayes theorem; not everyone is a Bayesian.

The inverse probability of Laplace and Gauss was right about the mathematics and imprecise about the philosophy. Fisher was right about the philosophy's difficulties and perhaps too hasty in concluding that the mathematics could be discarded. The modern synthesis — using the likelihood for what it can do and the posterior for what it can do, without pretending either is complete — is the position this book tries to articulate.

Exercise 4.16. Gauss argued that the Gaussian error distribution is the unique distribution for which the arithmetic mean is the posterior mode.

1. Verify this for $n = 2$ observations: show that $\frac{d}{d\theta}[\log f(y_1 - \theta) + \log f(y_2 - \theta)] = 0$ at $\theta = (y_1 + y_2)/2$ implies $f'(\varepsilon)/f(\varepsilon) \propto \varepsilon$, and conclude that $\log f(\varepsilon) \propto -\varepsilon^2$.

2. Gauss's argument assumes the posterior mode equals the arithmetic mean for *every* dataset. Why is this assumption necessary? What goes wrong if the mode equals the mean only on average?

3. In modern language, Gauss's derivation shows that least squares is MAP estimation under a Gaussian likelihood and a uniform prior. How does this connect to the discussion of ridge regression as MAP estimation in Chapter 8?

Exercise 4.17. Fisher's MLE and Laplace's posterior mode under a uniform prior are identical when the parameter space is unbounded. They differ when:

(a) The parameter is bounded (e.g., $\theta \in [0, 1]$). For $y \sim \text{Binomial}(n, \theta)$ with $k = 0$ successes, the MLE is $\hat{\theta} = 0$ but Laplace's posterior mode under a uniform prior is also 0, while the posterior mean is $1/(n + 2)$. Why does Laplace prefer the posterior mean over the mode in this case?

(b) The prior is not uniform. For $y \sim \mathcal{N}(\theta, 1)$ with prior $\theta \sim \mathcal{N}(0, \tau^2)$, show that the MAP estimate is $\hat{\theta}_{\text{MAP}} = y/(1 + 1/\tau^2)$, which differs from the MLE y. What does this correspond to in the regression context?

Chapter 5

Decision Theory and the Justification of Priors

The previous chapter introduced Bayesian inference as a procedure for updating beliefs: the prior encodes what the observer knows before seeing data, the likelihood encodes what the data says, and the posterior encodes the combination. But what should the observer *do* with the posterior? And is the prior merely a personal preference, or does it have a deeper justification that even a committed frequentist must accept?

Decision theory answers both questions. It provides a rigorous framework for turning a posterior distribution into an action — an estimate, a decision, a prediction — and it shows that the prior is not optional. Any observer who wants to compare estimators by a reasonable criterion must supply a weight function over the parameter space. That weight function is the prior, whether or not the observer chooses to call it one.

5.1 The Decision Problem

A **statistical decision problem** has three ingredients:

- A parameter space Θ: the set of possible values of the unknown θ.
- A data space $\mathcal{Y}$: the set of possible observations y.
- An action space $\mathcal{A}$: the set of possible actions a the observer can take. For estimation, $\mathcal{A} = \Theta$: the action is a guess for θ. For classification, $\mathcal{A} = \{0, 1, \ldots, K\}$: the action is a class label. For hypothesis testing, $\mathcal{A} = \{\text{reject}, \text{accept}\}$.

An **estimator** (or **decision rule**) is a function $\delta : \mathcal{Y} \to \mathcal{A}$ that maps observed data to an action.

Definition 5.1 (Loss Function)**.** A **loss function** $L : \Theta \times \mathcal{A} \to [0, \infty)$ measures the cost of taking action a when the true parameter is θ. The loss is zero when the action is perfect and positive otherwise.

The loss is the fundamental criterion of quality. Everything else — risk, Bayes risk, minimax — is built from it. Common choices:

- **Squared error loss:** $L(\theta, a) = (\theta - a)^2$. Penalizes large errors heavily. Appropriate when errors in either direction are equally costly and large errors are especially undesirable.
- **Absolute error loss:** $L(\theta, a) = |\theta - a|$. Penalizes all errors linearly. Robust to outliers compared to squared error.
- **0–1 loss:** $L(\theta, a) = \mathbf{1}[\theta \neq a]$. Zero for a correct decision, one for any wrong decision. The natural loss for classification.
- **Asymmetric loss:** $L(\theta, a) = c_1(\theta - a)^+ + c_2(a - \theta)^+$ for constants $c_1 \neq c_2$. Appropriate when overestimates and underestimates have different costs (e.g., medical dosing).

Definition 5.2 (Risk)**.** The **risk** of an estimator δ at parameter value θ is the expected loss:

$$R(\theta, \delta) = \mathbb{E}_y[L(\theta, \delta(y))] = \int L(\theta, \delta(y))\, p(y \mid \theta)\, dy$$

The expectation is over repeated observations y from the model $p(y \mid \theta)$, holding θ fixed. The risk is a function of θ: it measures how well the estimator δ performs when the truth is θ.

5.2 The Comparison Problem

With two estimators δ_1 and δ_2 in hand, how do we decide which is better? Their risks $R(\theta, \delta_1)$ and $R(\theta, \delta_2)$ are both functions of θ. In almost every non-trivial problem, these functions cross: δ_1 has lower risk for some values of θ and δ_2 for others. There is no universally better estimator.

Example 5.3 (Risk functions cross)**.** Suppose $y \sim \mathcal{N}(\theta, 1)$ and we compare two estimators under squared error loss:

- $\delta_1(y) = y$: the sample observation itself (the MLE). Risk: $R(\theta, \delta_1) = \mathbb{E}[(\theta - y)^2] = 1$ for all θ.
- $\delta_2(y) = 0$: always guess zero, regardless of data. Risk: $R(\theta, \delta_2) = \mathbb{E}[(\theta - 0)^2] = \theta^2$.

For $|\theta| < 1$, the constant estimator δ_2 has lower risk than δ_1. For $|\theta| > 1$, δ_1 is better. Neither dominates the other. If we believe θ is near zero, δ_2 is preferable; if θ could be large, δ_1 is safer. The comparison requires a judgment about where θ is likely to be — it requires a prior.

Definition 5.4 (Dominance and Admissibility)**.** An estimator δ_1 **dominates** δ_2 if:

$$R(\theta, \delta_1) \leq R(\theta, \delta_2) \quad \text{for all } \theta \in \Theta$$

with strict inequality for at least one θ. An estimator is **inadmissible** if it is dominated by some other estimator, and **admissible** if no such dominating estimator exists.

Inadmissible estimators should never be used: there exists another estimator that is at least as good everywhere and strictly better somewhere. Admissibility is a minimal sanity check — a necessary condition for a reasonable estimator, though not sufficient.

5.3 A Weight Function Must Appear

To reduce two risk functions to a single number for comparison, we must aggregate risk over the parameter space. The only mathematically natural way to do this is to integrate with a weight function.

Definition 5.5 (Bayes Risk)**.** Given a weight function $w(\theta) \geq 0$ with $\int w(\theta)\, d\theta = 1$ (so w is a probability distribution over Θ), the **Bayes risk** of an estimator δ under w is:

$$r(w, \delta) = \int R(\theta, \delta)\, w(\theta)\, d\theta = \int\int L(\theta, \delta(y))\, p(y \mid \theta)\, w(\theta)\, dy\, d\theta$$

The Bayes risk is a single number: the average risk, weighted by how likely each θ is according to w.

The weight function is inevitable. Any criterion that reduces the risk function $R(\theta, \delta)$ to a single number for comparison must integrate over θ with some weight. There is no weight-free way to compare estimators globally. The weight function $w(\theta)$ is the prior, whether it is called that or not. The question is not whether to have a prior, but whether to be explicit and thoughtful about it.

To see why no weight-free comparison is possible, suppose we try to rank estimators by their maximum risk instead. This is the minimax criterion (developed below). But as we will show, the minimax estimator is itself a Bayes estimator under a specific, adversarially chosen prior. Even the most aggressively prior-free frequentist approach secretly uses one.

5.4 The Bayes Estimator Under Squared Error Loss

The **Bayes estimator** under weight w is the estimator that minimizes the Bayes risk:

$$\delta_w = \arg\min_{\delta}\, r(w, \delta)$$

For squared error loss, there is an explicit formula.

Theorem 5.6 (Posterior Mean is Optimal Under Squared Error Loss)**.** *Under squared error loss $L(\theta, a) = (\theta - a)^2$, the Bayes estimator under prior w is the* ***posterior mean****:*

$$\delta_w(y) = \mathbb{E}_w[\theta \mid y] = \int \theta \, p(\theta \mid y) \, d\theta$$

Proof. We minimize the Bayes risk by minimizing it pointwise for each observed y. The Bayes risk decomposes as:

$$\begin{aligned} r(w, \delta) &= \int \int (\theta - \delta(y))^2 \, p(y \mid \theta) \, w(\theta) \, d\theta \, dy \\ &= \int \int (\theta - \delta(y))^2 \, p(\theta, y) \, d\theta \, dy \\ &= \int \underbrace{\left[\int (\theta - \delta(y))^2 \, p(\theta \mid y) \, d\theta \right]}_{\text{posterior expected loss at } y} p(y) \, dy \end{aligned}$$

Since $p(y) \geq 0$, we minimize the total by minimizing the bracketed term for each y separately. For fixed y, we minimize:

$$\int (\theta - a)^2 \, p(\theta \mid y) \, d\theta$$

over the action $a = \delta(y)$. Expand the square:

$$\begin{aligned} \int (\theta - a)^2 \, p(\theta \mid y) \, d\theta &= \int (\theta^2 - 2a\theta + a^2) \, p(\theta \mid y) \, d\theta \\ &= \mathbb{E}[\theta^2 \mid y] - 2a \, \mathbb{E}[\theta \mid y] + a^2 \end{aligned}$$

This is a quadratic in a, minimized by setting the derivative to zero:

$$\frac{d}{da} \left(\mathbb{E}[\theta^2 \mid y] - 2a \, \mathbb{E}[\theta \mid y] + a^2 \right) = -2\mathbb{E}[\theta \mid y] + 2a = 0$$

$$\Rightarrow \quad a^* = \mathbb{E}[\theta \mid y]$$

The minimizing action is the posterior mean. Since this holds for every y, the Bayes estimator is $\delta_w(y) = \mathbb{E}[\theta \mid y]$. □

Decomposing the posterior expected loss. The minimum value of $\int (\theta - a)^2 \, p(\theta \mid y) \, d\theta$ at $a = \mathbb{E}[\theta \mid y]$ is:

$$\mathbb{E}[\theta^2 \mid y] - (\mathbb{E}[\theta \mid y])^2 = \mathrm{Var}(\theta \mid y)$$

The posterior variance is the irreducible uncertainty about θ after observing y. No estimator can achieve expected squared error below $\mathrm{Var}(\theta \mid y)$ on average — this is the fundamental lower bound on estimation error. The posterior mean achieves this bound.

Example 5.7 (Gaussian posterior mean)**.** In the speed of light problem with prior $\theta \sim \mathcal{N}(\mu_0, \tau^2)$ and n measurements with noise σ^2, the posterior is $\mathcal{N}(\mu_n, \tau_n^2)$. The Bayes estimator under squared error loss is the posterior mean $\mu_n = \frac{(1/\tau^2)\mu_0 + (n/\sigma^2)\bar{y}}{1/\tau^2 + n/\sigma^2}$. This is the precision-weighted average of the prior mean and the sample mean — neither the MLE $\bar{y}$ alone nor the prior mean alone, but a combination that extracts the most information from both sources.

The bias-variance-prior tradeoff. The risk of the posterior mean $\delta_w(y) = \mathbb{E}[\theta \mid y]$ under squared error can be decomposed. For any estimator $\delta(y)$, write the expected squared error as:

$$R(\theta, \delta) = \mathrm{Var}_y(\delta(y)) + (\mathbb{E}_y[\delta(y)] - \theta)^2$$

The first term is the **variance** of the estimator (how much it fluctuates across datasets); the second is the squared **bias** (how far its average is from the truth). The MLE has zero bias but typically high variance. The posterior mean accepts some bias (it is pulled toward the prior mean) in exchange for reduced variance. The optimal tradeoff is exactly determined by the precisions of the prior and the data.

5.5 Optimal Estimation Under 0–1 Loss: Posterior Mode

Squared error loss is natural for continuous parameters. When the parameter is discrete — a class label, a hypothesis, a category — the natural loss is the **0–1 loss**: you pay 1 for any wrong answer and 0 for a correct one.

Theorem 5.8 (Posterior Mode is Optimal Under 0–1 Loss)**.** *Under 0–1 loss $L(\theta, a) = \mathbf{1}[\theta \neq a]$, the Bayes estimator under prior w is the* ***posterior mode*** *(the MAP estimate):*

$$\delta_w(y) = \arg\max_{\theta} \, p(\theta \mid y)$$

Proof. For fixed y, the posterior expected loss is:

$$\int \mathbf{1}[\theta \neq a] \, p(\theta \mid y) \, d\theta = P(\theta \neq a \mid y) = 1 - P(\theta = a \mid y) = 1 - p(a \mid y)$$

This is minimized by maximizing $p(a \mid y)$ over a, giving $a^* = \arg\max_a p(a \mid y)$, the posterior mode. □

Loss function determines optimal action:

Loss function	Optimal action	Name
Squared error $(\theta - a)^2$	$\mathbb{E}[\theta \mid y]$	Posterior mean
Absolute error $\lvert\theta - a\rvert$	$\mathrm{median}(\theta \mid y)$	Posterior median
0–1 loss $\mathbf{1}[\theta \neq a]$	$\arg\max_\theta \, p(\theta \mid y)$	Posterior mode (MAP)

The posterior distribution is the complete summary of knowledge about θ after observing y. The optimal point estimate depends on the loss function: what you report should depend on what it costs to be wrong.

An important special case: binary classification. Let $\theta \in \{0, 1\}$ be a class label and y be observed features. The posterior mode is:

$$\delta(y) = \begin{cases} 1 & \text{if } p(\theta = 1 \mid y) > 1/2 \\ 0 & \text{otherwise} \end{cases}$$

This is the **Bayes classifier**: classify to the more probable class. Under 0–1 loss, this minimizes the probability of error. No classifier achieves a lower error rate than the Bayes classifier, given the true posterior — this is the fundamental lower bound on classification error.

Asymmetric 0–1 loss. If false positives and false negatives have different costs — cost c_+ for calling 0 when truth is 1, and cost c_- for calling 1 when truth is 0 — the optimal decision rule shifts the threshold:

$$\delta(y) = 1 \quad \Leftrightarrow \quad p(\theta = 1 \mid y) > \frac{c_-}{c_+ + c_-}$$

When false negatives are more costly than false positives ($c_+ > c_-$), the threshold falls below 1/2: we classify as positive more liberally. Medical diagnosis is the canonical example: failing to detect a disease (false negative) may be far more costly than an unnecessary follow-up test (false positive).

5.6 The Weight Function and the ABC Prior

The weight function $w(\theta)$ in the Bayes risk has a precise operational meaning that connects directly to the ABC procedure of the previous chapter.

Recall ABC: to sample from the posterior, we draw candidate values $\theta^{(s)}$ from the prior $w(\theta)$ and keep those for which the simulated data matches the observed data. The prior is the distribution over which we *try* candidate values.

Now consider the Bayes risk from this perspective:

$$r(w, \delta) = \int\int (\theta - \delta(y))^2 \, p(y \mid \theta) \, w(\theta) \, d\theta \, dy$$

This is the expected squared error when we draw (θ, y) jointly from the **generative process**:

1. Draw a candidate $\theta \sim w(\theta)$ (the prior try).

2. Generate data $y \sim p(y \mid \theta)$ (the model).

3. Compute the loss $(\theta - \delta(y))^2$.

The Bayes risk is exactly the average loss over this generative loop. Minimizing $r(w, \delta)$ means finding the estimator that performs best on average when θ is drawn from w and data from the model. The weight function w determines how often each θ is tried.

The weight function as a trying distribution. The Bayes risk $r(w, \delta)$ measures how well δ performs when:

- Candidate parameters are tried according to $w(\theta)$.
- For each candidate, data is generated from $p(y \mid \theta)$.
- The cost of each trial is $L(\theta, \delta(y))$.

Minimizing $r(w, \delta)$ gives the estimator that is best on average under this trying distribution. The optimal estimator is the posterior mean (for squared error) or mode (for 0–1 loss) under the posterior induced by w. The ABC procedure and the Bayes estimator are two sides of the same coin: ABC generates samples from the joint, the Bayes estimator aggregates them optimally.

Different weight functions represent different priorities. Suppose a medical device manufacturer must estimate the sensitivity θ of a diagnostic test. They care more about accuracy in the range $\theta \in [0.8, 1.0]$ — clinically relevant values — than about accuracy near $\theta = 0$. Their weight function w should reflect this: it should put most of its mass in $[0.8, 1.0]$. The resulting Bayes estimator will sacrifice accuracy for very small θ in exchange for greater accuracy where it matters. This is not dishonest — it is a principled statement of what the estimator should optimize for.

5.7 Minimax Estimation

What if an observer refuses to specify any weight function? The most common frequentist response is the **minimax criterion**: choose the estimator that minimizes the worst-case risk.

Definition 5.9 (Minimax Estimator)**.** The **minimax estimator** δ_{mm} minimizes the maximum risk over all θ:

$$\delta_{\text{mm}} = \arg\min_{\delta} \sup_{\theta \in \Theta} R(\theta, \delta)$$

The **minimax risk** is $V^* = \min_\delta \sup_\theta R(\theta, \delta)$.

The minimax estimator is the optimal strategy in the worst case: it is the best choice if nature is adversarially choosing θ to maximize our error. In game theory terms, it is the optimal strategy in a two-player zero-sum game between the statistician (choosing δ) and nature (choosing θ).

The Minimax Theorem

Computing the minimax estimator directly from the definition — minimizing over all decision rules of the maximum over all θ — is generally intractable. The following theorem provides a fundamental shortcut.

Theorem 5.10 (Minimax = Bayes under Least Favorable Prior)**.** *Suppose δ_w is the Bayes estimator under prior w, and suppose the Bayes risk equals the maximum risk:*

$$r(w, \delta_w) = \sup_\theta R(\theta, \delta_w)$$

(i.e., the risk of δ_w is constant in θ). Then:

1. *δ_w is the minimax estimator.*
2. *w is the **least favorable prior**: the prior that maximizes the Bayes risk, $w = \arg\max_{w'} r(w', \delta_{w'})$.*

Proof. Let δ be any other estimator. Then:

$$\begin{aligned}\sup_\theta R(\theta, \delta) &\geq \int R(\theta, \delta)\, w(\theta)\, d\theta = r(w, \delta) \\ &\geq r(w, \delta_w) \qquad \text{(since } \delta_w \text{ minimizes Bayes risk under } w\text{)} \\ &= \sup_\theta R(\theta, \delta_w) \qquad \text{(by assumption)}\end{aligned}$$

So $\sup_\theta R(\theta, \delta) \geq \sup_\theta R(\theta, \delta_w)$ for any δ, which means δ_w achieves the minimum worst-case risk. □

The intuition. A prior w whose Bayes estimator has constant risk is called *least favorable* because it is the hardest prior for the statistician to cope with. If risk is constant across θ, then no matter what strategy the statistician uses, the worst θ is equally bad for all strategies. The statistician cannot exploit any structure in the risk function to do better.

The theorem says: to find the minimax estimator, find the prior that makes the Bayes estimator's risk as flat as possible. The flattest possible risk is achieved by the least favorable prior. This is the minmax theorem: the minimax estimator and the Bayes estimator under the least favorable prior are the same.

Example 5.11 (Minimax for Gaussian mean)**.** Let $y \sim \mathcal{N}(\theta, \sigma^2)$ with $\theta \in [-M, M]$ for some known M. We seek the minimax estimator under squared error loss.

Consider the Bayes estimator under the prior $\theta \sim \mathcal{N}(0, \tau^2)$:

$$\delta_\tau(y) = \mathbb{E}[\theta \mid y] = \frac{\tau^2}{\tau^2 + \sigma^2}\, y = (1 - B)\, y, \quad B = \frac{\sigma^2}{\tau^2 + \sigma^2}$$

This is a shrinkage estimator: it pulls y toward zero by factor $(1 - B)$. Its risk at parameter value θ is:

$$\begin{aligned} R(\theta, \delta_\tau) &= \mathbb{E}\big[(\theta - (1-B)y)^2\big] \\ &= \mathbb{E}\big[((1-B)(\theta - y) - B\theta)^2\big] \\ &= (1-B)^2\sigma^2 + B^2\theta^2 \end{aligned}$$

This risk is not constant in θ — it is a quadratic in θ, minimized at $\theta = 0$ and increasing toward the boundary $\pm M$. To make the risk flat, we choose τ^2 so that the risk at $\theta = M$ equals the Bayes risk:

$$r(\tau, \delta_\tau) = \int R(\theta, \delta_\tau)\, \mathcal{N}(\theta; 0, \tau^2)\, d\theta = (1-B)^2\sigma^2 + B^2\tau^2 = \frac{\sigma^2\tau^2}{\sigma^2 + \tau^2}$$

Setting the maximum risk equal to the Bayes risk — i.e., requiring the risk to be flat — gives $\tau^2 = M^2$ (to leading order for large M). The resulting Bayes estimator under $\mathcal{N}(0, M^2)$ is then the minimax estimator for $\theta \in [-M, M]$. As $M \to \infty$ (the unrestricted case), the least favorable prior spreads to a flat (improper) prior, and the minimax estimator approaches the MLE y.

The Minimax Estimator Has a Prior

The conclusion of the minimax theorem is striking: the observer who refuses to specify a prior, and instead optimizes for the worst case, has in fact chosen the most adversarial possible prior — the least favorable one. There is no prior-free frequentist procedure that achieves the minimax bound. The minimax estimator *is* a Bayes estimator; it just uses the most pessimistic prior imaginable.

No escape from the prior. Every reasonable estimator uses a weight function:

- A Bayesian uses the prior that encodes their genuine beliefs or prior knowledge.
- A minimax frequentist uses the least favorable prior that makes the worst case as small as possible.
- A practitioner who ignores the prior and uses the MLE is implicitly using a flat (uniform) weight over θ — a specific prior choice.

The difference between Bayesian and frequentist estimation is not whether a prior is used. It is whether the prior is chosen thoughtfully.

5.8 The Complete Class Theorem

The connection between Bayes and admissible estimators goes even deeper. The **complete class theorem** says that the class of Bayes estimators is, in a precise sense,

complete: it exhausts all the good estimators.

Theorem 5.12 (Complete Class Theorem, informal)**.** *Under mild regularity conditions, every admissible estimator is a Bayes estimator (or a limit of Bayes estimators) under some prior. Equivalently, if an estimator is not a Bayes estimator under any prior, then it is inadmissible: there exists another estimator that dominates it uniformly.*

The proof uses results from convex analysis and game theory and is beyond our scope. But the implication is decisive: if you want to use an admissible estimator — one that is not beaten everywhere by something else — you must look among the Bayes estimators. The class of Bayes estimators contains all the good ones.

Example 5.13 (MLE is inadmissible in high dimensions)**.** The most striking application of the complete class theorem is the James-Stein result, which we examine in detail in the next chapter. Briefly: for estimating a d-dimensional Gaussian mean $\theta \in \mathbb{R}^d$ from a single observation $y \sim \mathcal{N}(\theta, I_d)$ under squared error loss, the MLE $\delta_{\text{MLE}}(y) = y$ is inadmissible for $d \geq 3$. The James-Stein estimator $\delta_{\text{JS}}(y) = (1 - (d-2)/\|y\|^2)y$ uniformly dominates the MLE. And δ_{JS} is (approximately) the Bayes estimator under a Gaussian prior $\theta \sim \mathcal{N}(0, \tau^2 I_d)$ for an appropriate τ^2. The MLE's inadmissibility is a consequence of its not being a Bayes estimator under any reasonable prior.

5.9 A Complete Example: Estimating a Bias

We work through a complete decision-theoretic analysis to tie the pieces together.

Setup. A coin is flipped $n = 20$ times, yielding $k = 14$ heads. The unknown bias is $\theta \in [0, 1]$. We consider three estimators:

- $\delta_{\text{MLE}}(k) = k/n$: the MLE.
- $\delta_{\text{Bayes}}(k)$: the Bayes estimator under $\theta \sim \text{Beta}(\alpha, \alpha)$ for some $\alpha > 0$.
- $\delta_{\text{mm}}(k)$: the minimax estimator.

The MLE. $\delta_{\text{MLE}}(k) = k/n = 14/20 = 0.70$. Risk under squared error:

$$R(\theta, \delta_{\text{MLE}}) = \text{Var}(k/n) = \frac{\theta(1-\theta)}{n}$$

Maximum risk over $\theta \in [0, 1]$ is at $\theta = 1/2$: $\sup_\theta R(\theta, \delta_{\text{MLE}}) = 1/(4n) = 1/80$.

The Bayes estimator. Under $\theta \sim \text{Beta}(\alpha, \alpha)$ (symmetric prior with mean $1/2$) and $k \sim \text{Binomial}(n, \theta)$, the posterior is $\text{Beta}(\alpha + k,\ \alpha + n - k)$. The posterior mean is:

$$\delta_\alpha(k) = \frac{\alpha + k}{2\alpha + n} = \frac{\alpha}{2\alpha + n} \cdot \frac{1}{2} + \frac{n}{2\alpha + n} \cdot \frac{k}{n}$$

This is a weighted average of the prior mean $1/2$ and the MLE k/n, with weight $n/(2\alpha + n)$ on the data. The risk is:

$$R(\theta, \delta_\alpha) = \text{Var}(\delta_\alpha(k)) + \text{Bias}^2(\delta_\alpha(k))$$

After calculation:

$$R(\theta, \delta_\alpha) = \frac{n\theta(1-\theta)}{(2\alpha+n)^2} + \frac{\alpha^2(2\theta-1)^2}{(2\alpha+n)^2}$$

The minimax estimator. To make the risk constant in θ, we need:

$$\frac{n\theta(1-\theta)}{(2\alpha+n)^2} + \frac{\alpha^2(2\theta-1)^2}{(2\alpha+n)^2} = C$$

for some constant C. Expanding $(2\theta-1)^2 = 4\theta^2 - 4\theta + 1$ and $\theta(1-\theta) = \theta - \theta^2$:

$$\begin{aligned}\text{Numerator} &= n(\theta - \theta^2) + \alpha^2(4\theta^2 - 4\theta + 1) \\ &= (4\alpha^2 - n)\theta^2 + (n - 4\alpha^2)\theta + \alpha^2\end{aligned}$$

For this to be constant in θ, the coefficients of θ^2 and θ must vanish:

$$4\alpha^2 - n = 0 \quad \Rightarrow \quad \alpha = \frac{\sqrt{n}}{2}$$

With $n = 20$: $\alpha = \sqrt{20}/2 \approx 2.24$. The minimax estimator is the Bayes estimator under Beta$(\sqrt{n}/2, \sqrt{n}/2)$:

$$\delta_{\text{mm}}(k) = \frac{\sqrt{n}/2 + k}{\sqrt{n} + n}$$

For our data ($n = 20$, $k = 14$):

$$\delta_{\text{mm}}(14) = \frac{2.24 + 14}{\sqrt{20} + 20} = \frac{16.24}{24.47} \approx 0.664$$

The constant minimax risk is:

$$C = \frac{\alpha^2}{(2\alpha+n)^2} = \frac{n/4}{(\sqrt{n}+n)^2} = \frac{1}{4(\sqrt{n}+1)^2/\sqrt{n}\cdot\sqrt{n}} \approx \frac{1}{4(1+\sqrt{n})^2}$$

For $n = 20$: $C = 1/(4(1+\sqrt{20})^2) \approx 1/107$, which is less than the MLE's maximum risk of $1/80$. The minimax estimator improves on the MLE in the worst case.

Estimator	Formula	$\delta(k{=}14)$	$\sup R(\theta, \delta)$	At θ
MLE	k/n	0.700	$1/80 = 0.0125$	$\theta = 1/2$
Bayes, $\alpha = 1$	$(1+k)/(2+n)$	0.682	0.0104	$\theta = 0, 1$
Minimax ($\alpha = \sqrt{n}/2$)	$(\sqrt{n}/2 + k)/(\sqrt{n}+n)$	0.664	≈ 0.0093	all θ

The minimax estimator shrinks the estimate toward 1/2 compared to the MLE. It sacrifices some accuracy near $\theta = 1/2$ (where the MLE is already good) to gain accuracy near $\theta = 0$ and $\theta = 1$ (where the MLE's risk spikes). The prior $\text{Beta}(\sqrt{n}/2, \sqrt{n}/2)$ is the least favorable prior: the prior under which estimation is hardest.

Exercise 5.14. Under absolute error loss $L(\theta, a) = |\theta - a|$, show that the Bayes optimal estimator is the posterior median. (Hint: differentiate $\int |\theta - a|\, p(\theta \mid y)\, d\theta$ with respect to a and set the derivative to zero, using the fact that $d|\theta - a|/da = -1$ for $\theta > a$ and $+1$ for $\theta < a$.)

Exercise 5.15. A classifier must decide whether an email is spam ($\theta = 1$) or not spam ($\theta = 0$). The posterior probability of spam given the email's features is $p(\theta = 1 \mid y) = 0.3$. Under symmetric 0–1 loss, what is the optimal decision? Now suppose the cost of missing a spam (false negative) is twice the cost of a false alarm (false positive). What is the new decision threshold, and what is the optimal decision at $p(\theta = 1 \mid y) = 0.3$?

Exercise 5.16. For the coin-flipping example with $n = 20$ and $k = 14$:

1. Compute the posterior mean and posterior 95% credible interval under $\text{Beta}(1, 1)$, $\text{Beta}(2, 2)$, and $\text{Beta}(\sqrt{20}/2, \sqrt{20}/2)$ priors.

2. Plot the risk functions $R(\theta, \delta_\alpha)$ for $\alpha = 1$, $\alpha = 2$, and $\alpha = \sqrt{n}/2$ on the same axes, along with $R(\theta, \delta_{\text{MLE}})$. Verify visually that the minimax risk function is flat.

3. For each prior, compute the Bayes risk $r(w, \delta_\alpha)$ and verify that it equals the constant minimax risk when $\alpha = \sqrt{n}/2$.

5.10 What the Weight Function Represents: Various Views

We have now seen the weight function appear in three guises. It is worth pausing to compare them.

View 1: Subjective belief (de Finetti). The weight function $w(\theta)$ represents the observer's personal probability over θ before seeing data. This is the standard Bayesian interpretation. The prior is subjective in the sense that different observers may have different priors, but it is not arbitrary: it must be coherent (consistent with the axioms of probability) and it must reflect genuine prior knowledge.

View 2: Design criterion (pragmatic). The weight function specifies which values of θ the analyst cares most about estimating accurately. A practitioner who builds a model primarily for $\theta \in [0.5, 1.0]$ uses a prior concentrated in that range. The resulting Bayes estimator sacrifices performance outside $[0.5, 1.0]$ for better performance inside. The prior is a design choice, not a statement of belief.

View 3: Trying distribution (ABC). As developed in the previous chapter and above, the weight function describes how candidate values are tried in the ABC loop. It is the sampling distribution for the parameter search. A uniform prior tries all candidate values equally; a Gaussian prior tries values near μ_0 more often. The posterior is the conditional distribution of the tried value, given that it generated data matching the observation.

View 4: Adversarial prior (minimax). The least favorable prior is the adversary's optimal strategy in the minimax game. It concentrates weight on the hardest values of θ to estimate — the values where no estimator does well.

All four views lead to the same mathematics. The differences are interpretational, not computational. Once w is fixed, the Bayes estimator is the posterior mean (for squared error) or mode (for 0–1 loss), regardless of why w was chosen. The framework is robust to disagreements about the meaning of probability: a subjective Bayesian, a pragmatic engineer, an ABC practitioner, and a minimax frequentist all compute the same posterior once they agree on w.

5.11 Why Weight Function and ABC Arguments Are More Convincing

The classical justifications for Bayesian priors fall into three families: Cox's theorem, Dutch book arguments, and Savage's axioms. Each is mathematically elegant. Each has also been contested for decades, and none has succeeded in converting skeptics. It is worth understanding why, and why the decision-theoretic and ABC arguments developed in this book make a stronger case — not merely that you should have a prior, but that specific situations have specific right answers for what the prior should be.

The Classical Arguments and Their Limitations

Cox's theorem (1946) shows that any system of plausible reasoning that satisfies a small set of desiderata — consistency, universality, and continuity — must be isomorphic to probability theory. If you want to reason consistently about uncertainty, you are forced to use probabilities, and Bayes rule follows from the product rule. The difficulty is in the desiderata themselves. Cox's conditions are sufficiency conditions reverse-engineered from the conclusion. A skeptic can always ask: why must plausible reasoning be continuous? Why universal? And even granting the conclusion, Cox's theorem says nothing about which prior to use. It establishes the framework and then stops.

Dutch book arguments (de Finetti, Ramsey) show that an agent whose probabilities violate the axioms can be made to accept a collection of bets guaranteeing a loss. This establishes coherence as a minimum requirement. Its limitation is the same:

it tells you that your prior must be a probability measure, but nothing about which one. The Dutch book argument is a consistency requirement, not a construction. And the premise — that beliefs should be operationalized as betting odds — is itself contestable. Scientists assigning probabilities to physical constants are not placing bets. The framework does not match how they think.

Savage's axioms (1954) provide the most complete foundation: behavioral axioms about preferences among acts that together imply the existence of a subjective probability and utility function. Savage's theorem is deep mathematics. Its limitation is that the axioms describe an idealized rational agent. The sure-thing principle is violated by the Allais paradox and many other empirically documented patterns of choice. And again: Savage's theorem tells you that a prior exists; it does not tell you what it should be.

All three classical arguments share a common structure. They are conditional: *if* you want to reason consistently, *if* you want to avoid being exploited, *if* you satisfy behavioral axioms, then you must reason like a Bayesian. They establish the necessity of priors. But the practicing statistician's real question is not "must I have a prior?" It is "what prior should I use?" On this question, the classical arguments are silent.

Our Arguments Answer the Harder Question

The decision-theoretic and ABC arguments do not merely tell you that a prior must exist. They tell you what it should be, in terms you can act on.

The ABC argument prescribes the prior directly. The prior is your search strategy over the parameter space: the distribution over which you try candidate values before filtering through the data. This prescription is constructive and specific.

What you know, you should encode. If previous physical experiments suggest the speed of light is near $\mu_0 = 299{,}800$ km/s with uncertainty $\tau = 100$ km/s, then $\theta \sim \mathcal{N}(\mu_0, \tau^2)$ is not a philosophical stance. It is a description of where the ABC loop should try more densely. The search should concentrate where the truth is more likely to be, because that makes the filter efficient. A prior that ignores this knowledge — say, a uniform prior over all positive reals — would waste most of its tries on values that physical theory already rules out. Encoding prior knowledge in the prior is not optional if you want an efficient search. It is the rational strategy.

What you do not know, you should not pretend to. Before seeing Laplace's birth data, there is no reason to prefer any value of the birth probability p over any other. The ABC loop should therefore try values uniformly across $[0, 1]$. The uniform prior $\text{Beta}(1, 1)$ is not a statement of ignorance as a philosophical position; it is the honest search strategy of an observer who has no information and therefore no reason to try some values more than others.

Jeffreys prior is the right search strategy for a parameterization-free observer. If you want to search without favoring any parameterization — so that your strategy for trying values of θ is the same as your strategy for trying values of $g(\theta)$ for any monotone g — the unique answer is the Jeffreys prior $p(\theta) \propto \sqrt{I(\theta)}$. This is not derived from

axioms about rationality. It is derived from the requirement that the search strategy be invariant to how you label the unknowns. The prior is determined by the geometry of the problem.

The decision-theoretic argument prescribes the prior from what you care about. The Bayes risk $r(w, \delta) = \int R(\theta, \delta)\, w(\theta)\, d\theta$ is the average risk when θ is drawn from w. Minimizing the Bayes risk gives the optimal estimator for an observer whose goal is to do well on average over the distribution w. This means:

The prior should reflect where accuracy matters. A medical device manufacturer who cares about performance for $\theta \in [0.8, 1.0]$ should use a prior concentrated there. The resulting Bayes estimator deliberately sacrifices accuracy for small θ — where the device will never operate — to gain accuracy where it will. This is not a subjective belief; it is a design specification. The prior encodes the objective.

The least favorable prior tells you the hardest problem. When you do not know what you care about and want to perform well in the worst case, the minimax theorem prescribes the prior: it is the least favorable distribution, the one that makes the estimation problem as hard as possible. For the coin-flipping problem with n tosses, this is $\text{Beta}(\sqrt{n}/2, \sqrt{n}/2)$. This prior is not chosen by introspection. It is derived by solving an equation: find α such that the Bayes estimator's risk is constant in θ. The worst-case-optimal strategy is a specific answer to a specific equation.

The James-Stein prior tells you the right amount of shrinkage. In the d-dimensional Gaussian mean problem, the empirical Bayes argument determines the prior from the data: $\theta_i \sim \mathcal{N}(0, \tau^2)$ with $\hat{\tau}^2 = \|y\|^2/d - \sigma^2$. This is not a belief about the θ_i; it is the prior whose Bayes estimator is the James-Stein estimator — the one that minimizes total squared error. The right amount of shrinkage is determined by the ratio of signal to noise, estimated from the data. The prior is a parameter of the estimator, fitted from evidence.

The Prior Is Determined, Not Assumed

The deeper point is this. The classical arguments ask you to accept a prior on faith — or on abstract principles of coherence that may or may not resonate. Our arguments say something more specific: in each situation, the right prior is determined by the structure of the problem. It is not a free parameter to be chosen arbitrarily. It is the answer to a precise question:

Question	Answer (the prior)
What do I know before seeing data?	Encode that knowledge as a density: Gaussian, Beta, or whatever family fits.
What do I not know?	Distribute tries evenly: the uniform or Jeffreys prior, depending on parameterization.
Where does accuracy matter?	Concentrate the prior there: the Bayes estimator will optimize for that region.
What is the worst case?	Solve for the least favorable prior: it is determined by the model and the loss function.
How much should I shrink?	Estimate the prior variance from the data: empirical Bayes gives the James-Stein answer.

The complete case for Bayesian priors.

1. **The prior is inevitable** (decision theory). Any scalar criterion for comparing estimators requires a weight function over the parameter space. This is arithmetic. Calling it a prior is simply being honest.

2. **The prior is a search strategy** (ABC). It describes where to try candidate values before filtering through the data. This makes its construction concrete: encode what you know, distribute evenly what you do not, be parameterization-invariant if that is your goal.

3. **The prior is determined by the problem** (both). Prior knowledge determines it from above; the loss function and the model determine it from below. In specific cases — Jeffreys, least favorable, empirical Bayes — the prior has a unique correct answer derived from the problem structure, not from philosophical commitment.

4. **The prior matters less as data grows** (consistency). With enough data, all reasonable priors converge to the same posterior. The ABC filter becomes tight enough that the starting search strategy is overwhelmed by evidence. The prior is most influential when data is scarce — which is precisely when prior knowledge is most valuable and most deserving of careful thought.

This is the most honest and constructive justification for Bayesian statistics. It does not ask you to accept axioms of rationality or to operationalize beliefs as betting odds. It asks only: what do you know, what do you care about, and where do you want to

search? The answers to these questions determine the prior. Making the prior explicit and thoughtful is not a philosophical commitment. It is the natural consequence of asking, and answering, the right questions.

Chapter 6

Shrinkage, the Stein Estimator, and Regularization

The previous chapter established that the posterior mean is the optimal estimator under squared error loss, and that every admissible estimator is a Bayes estimator under some prior. This chapter makes those abstract results concrete by examining what happens when we try to estimate many parameters simultaneously. The answer is surprising, practically important, and one of the most instructive results in all of statistics: the natural estimator — just use the data directly — is provably suboptimal in high dimensions, and the fix is Bayesian shrinkage.

6.1 Estimating a Single Gaussian Mean

We begin with the simplest possible estimation problem. Observe a single measurement:

$$y \sim \mathcal{N}(\theta, \sigma^2)$$

where $\theta \in \mathbb{R}$ is unknown and σ^2 is known. The maximum likelihood estimator (MLE) is $\hat{\theta}_{\text{MLE}} = y$. Its risk under squared error loss is:

$$R(\theta, \hat{\theta}_{\text{MLE}}) = \mathbb{E}[(\theta - y)^2] = \sigma^2$$

This is constant in θ: the MLE is equally good (or bad) everywhere. It is also the minimax estimator: from the previous chapter, the least favorable prior for a Gaussian mean with no bound on θ is the flat prior, giving the MLE. And it is admissible: no estimator has uniformly lower risk in one dimension.

The MLE is, in every sense, the right answer for $d = 1$.

Now suppose we want to estimate n measurements simultaneously. Each $y_i \sim \mathcal{N}(\theta_i, \sigma^2)$ independently, for $i = 1, \ldots, n$. The vector of observations is $y = (y_1, \ldots, y_n)$, and the vector of unknowns is $\theta = (\theta_1, \ldots, \theta_n)$. The MLE is $\hat{\theta}_{\text{MLE}} = y$, and the total

risk under squared error loss is:

$$R(\theta, \hat{\theta}_{\text{MLE}}) = \mathbb{E}[\|\theta - y\|^2] = \sum_{i=1}^{n} \mathbb{E}[(\theta_i - y_i)^2] = n\sigma^2$$

This scales linearly with n. Each coordinate is estimated independently, and there is no sharing of information across coordinates. For $d = 1$ and $d = 2$, this is optimal. For $d \geq 3$, it is not. This is Stein's theorem, and it is one of the most shocking results in statistics.

6.2 Stein's Theorem

Theorem 6.1 (Stein, 1956)**.** *Let $y \sim \mathcal{N}(\theta, \sigma^2 I_d)$ with $d \geq 3$. The MLE $\hat{\theta}_{\text{MLE}} = y$ is **inadmissible** under squared error loss. It is dominated by the **James-Stein estimator**:*

$$\hat{\theta}_{\text{JS}} = \left(1 - \frac{(d-2)\sigma^2}{\|y\|^2}\right) y$$

which satisfies:

$$R(\theta, \hat{\theta}_{\text{JS}}) < R(\theta, \hat{\theta}_{\text{MLE}}) = d\sigma^2$$

for every $\theta \in \mathbb{R}^d$.

Before proving this, we need to understand why it is so surprising.

Why this seems impossible. The d coordinates $y_1, \ldots, y_d$ are independent. Knowing y_1 tells you nothing about $\theta_2, \ldots, \theta_d$. If you are trying to estimate the batting averages of d baseball players, the population of d cities, or the effects of d unrelated treatments, each measured independently, it seems obvious that you should estimate each one from its own data. What could the batting average of one player possibly tell you about another?

Yet Stein showed that using all d measurements together — specifically, shrinking the entire vector y toward the origin — uniformly reduces the total squared error, for every θ. The coordinates are independent in the data, but they are connected in the estimator. And the connection is not a coincidence or a trick. It has a clean geometric explanation.

6.3 The Geometric Idea: Length Shrinkage

Stein's original insight was geometric. It concerns the *length* of the observed vector y.

The Length of y is Biased Upward

In the one million people picture, imagine one million copies of the experiment: one million draws of $y \sim \mathcal{N}(\theta, \sigma^2 I_d)$. Each draw is a random vector in $\mathbb{R}^d$. The true parameter θ is a fixed point in $\mathbb{R}^d$.

Each observation y is the true value θ plus noise: $y = \theta + \varepsilon$ where $\varepsilon \sim \mathcal{N}(0, \sigma^2 I_d)$. The noise pushes y away from θ in a random direction. What does this do to the *length* $\|y\|$?

Expand:

$$\|y\|^2 = \|\theta + \varepsilon\|^2 = \|\theta\|^2 + 2\theta^\top \varepsilon + \|\varepsilon\|^2$$

Taking expectations: $\mathbb{E}[\theta^\top \varepsilon] = 0$ (noise is unbiased), and $\mathbb{E}[\|\varepsilon\|^2] = d\sigma^2$ (each of the d coordinates has variance σ^2). Therefore:

$$\mathbb{E}[\|y\|^2] = \|\theta\|^2 + d\sigma^2$$

The expected squared length of y *exceeds* the squared length of θ by $d\sigma^2$. The noise inflates the length. This inflation grows with d: in high dimensions, the noise contribution $d\sigma^2$ can far exceed the signal $\|\theta\|^2$.

The fundamental geometric fact. In d dimensions, the noise vector $\varepsilon \sim \mathcal{N}(0, \sigma^2 I_d)$ has expected squared length $d\sigma^2$. It is nearly perpendicular to θ (the signal and noise directions are independent), and its length concentrates around $\sqrt{d}\sigma$ for large d. The observed vector $y = \theta + \varepsilon$ therefore points in a direction rotated away from θ, and its length is inflated by the noise.

Correcting the Length

If $\|y\|^2$ overestimates $\|\theta\|^2$ by approximately $d\sigma^2$, a natural correction is to shrink y toward the origin. How much should we shrink?

We want an estimator of the form $\hat{\theta} = c \cdot y$ for some scalar $c \in (0, 1)$. The risk is:

$$\begin{aligned} R(\theta, c\,y) &= \mathbb{E}[\|cy - \theta\|^2] \\ &= \mathbb{E}[\|c(\theta + \varepsilon) - \theta\|^2] \\ &= \mathbb{E}[\|(c-1)\theta + c\varepsilon\|^2] \\ &= (c-1)^2\|\theta\|^2 + c^2 d\sigma^2 \end{aligned}$$

This is minimized over c by differentiating and setting to zero:

$$\frac{d}{dc}\left[(c-1)^2\|\theta\|^2 + c^2 d\sigma^2\right] = 2(c-1)\|\theta\|^2 + 2c\,d\sigma^2 = 0$$

$$\Rightarrow \quad c^* = \frac{\|\theta\|^2}{\|\theta\|^2 + d\sigma^2}$$

The optimal shrinkage factor is $c^* = \|\theta\|^2/(\|\theta\|^2 + d\sigma^2) < 1$. We should always shrink y toward the origin. But c^* depends on $\|\theta\|^2$, which is unknown.

Estimating the Shrinkage Factor

The James-Stein estimator replaces the unknown $c^* = \|\theta\|^2/(\|\theta\|^2 + d\sigma^2)$ with a data-driven approximation. Since $\mathbb{E}[\|y\|^2] = \|\theta\|^2 + d\sigma^2$, we have:

$$c^* = \frac{\|\theta\|^2}{\|\theta\|^2 + d\sigma^2} = 1 - \frac{d\sigma^2}{\|\theta\|^2 + d\sigma^2} \approx 1 - \frac{d\sigma^2}{\|y\|^2}$$

where we replaced $\|\theta\|^2 + d\sigma^2 = \mathbb{E}[\|y\|^2]$ by $\|y\|^2$. But simply using $d\sigma^2/\|y\|^2$ as the shrinkage amount slightly overshrinks. Stein showed by an elegant calculation (using integration by parts and the Gaussian score function) that the correct factor is $(d-2)\sigma^2/\|y\|^2$, giving:

$$\hat{\theta}_{\text{JS}} = \left(1 - \frac{(d-2)\sigma^2}{\|y\|^2}\right) y$$

The factor $(d-2)$ rather than d is the precise correction that makes the risk strictly less than $d\sigma^2$ for all θ.

Proof of Risk Reduction

We now prove that the James-Stein estimator strictly dominates the MLE. The key tool is Stein's identity for Gaussian random variables.

Lemma 6.2 (Stein's Identity)**.** *Let $y \sim \mathcal{N}(\theta, \sigma^2 I_d)$ and let $g : \mathbb{R}^d \to \mathbb{R}^d$ be a weakly differentiable function with $\mathbb{E}[\|\nabla \cdot g(y)\|] < \infty$. Then:*

$$\mathbb{E}\big[(y-\theta)^\top g(y)\big] = \sigma^2 \mathbb{E}[\nabla \cdot g(y)]$$

where $\nabla \cdot g = \sum_{i=1}^d \partial g_i/\partial y_i$ is the divergence of g.

Proof. For a single coordinate:

$$\begin{aligned}\mathbb{E}[(y_i - \theta_i) g_i(y)] &= \int (y_i - \theta_i) g_i(y) \frac{e^{-\|y-\theta\|^2/(2\sigma^2)}}{(2\pi\sigma^2)^{d/2}}\, dy \\ &= \sigma^2 \int \frac{\partial}{\partial y_i}\left[\frac{e^{-\|y-\theta\|^2/(2\sigma^2)}}{(2\pi\sigma^2)^{d/2}}\right] g_i(y)\, dy\end{aligned}$$

Integrating by parts in y_i (boundary terms vanish):

$$= \sigma^2 \int \frac{e^{-\|y-\theta\|^2/(2\sigma^2)}}{(2\pi\sigma^2)^{d/2}} \frac{\partial g_i}{\partial y_i}(y)\, dy = \sigma^2 \mathbb{E}\left[\frac{\partial g_i}{\partial y_i}(y)\right]$$

Summing over $i = 1, \ldots, d$ gives the result. □

Now write the James-Stein estimator as:

$$\hat{\theta}_{\text{JS}} = y + g(y), \quad g(y) = -\frac{(d-2)\sigma^2}{\|y\|^2}\, y$$

The risk of any estimator $y + g(y)$ is:

$$\begin{aligned} R(\theta, y+g) &= \mathbb{E}[\|y + g(y) - \theta\|^2] \\ &= \mathbb{E}[\|y-\theta\|^2] + 2\mathbb{E}[(y-\theta)^\top g(y)] + \mathbb{E}[\|g(y)\|^2] \\ &= d\sigma^2 + 2\sigma^2 \mathbb{E}[\nabla \cdot g(y)] + \mathbb{E}[\|g(y)\|^2] \end{aligned}$$

using Stein's identity in the second step. Now compute the two remaining terms.

Divergence of g. We have $g_i(y) = -(d-2)\sigma^2 y_i / \|y\|^2$:

$$\begin{aligned} \frac{\partial g_i}{\partial y_i} &= -(d-2)\sigma^2 \frac{\partial}{\partial y_i} \frac{y_i}{\|y\|^2} \\ &= -(d-2)\sigma^2 \left(\frac{1}{\|y\|^2} - \frac{2y_i^2}{\|y\|^4} \right) \end{aligned}$$

Summing over i:

$$\nabla \cdot g = -(d-2)\sigma^2 \left(\frac{d}{\|y\|^2} - \frac{2\|y\|^2}{\|y\|^4} \right) = -(d-2)\sigma^2 \cdot \frac{d-2}{\|y\|^2} = -\frac{(d-2)^2\sigma^2}{\|y\|^2}$$

Squared norm of g.

$$\|g(y)\|^2 = \frac{(d-2)^2\sigma^4}{\|y\|^4}\|y\|^2 = \frac{(d-2)^2\sigma^4}{\|y\|^2}$$

Combining:

$$\begin{aligned} R(\theta, \hat{\theta}_{\mathrm{JS}}) &= d\sigma^2 + 2\sigma^2 \cdot \mathbb{E}\left[-\frac{(d-2)^2\sigma^2}{\|y\|^2} \right] + \mathbb{E}\left[\frac{(d-2)^2\sigma^4}{\|y\|^2} \right] \\ &= d\sigma^2 - (d-2)^2\sigma^4 \, \mathbb{E}\left[\frac{1}{\|y\|^2} \right] \end{aligned}$$

Since $\mathbb{E}[1/\|y\|^2] > 0$ for $d \geq 3$ (the expectation is finite because $\|y\|^2 \sim \sigma^2 \chi^2_d$ and $\mathbb{E}[1/\chi^2_d] = 1/(d-2)$ for $d > 2$):

$$R(\theta, \hat{\theta}_{\mathrm{JS}}) = d\sigma^2 - (d-2)^2\sigma^4 \, \mathbb{E}\left[\frac{1}{\|y\|^2} \right] < d\sigma^2 = R(\theta, \hat{\theta}_{\mathrm{MLE}})$$

for all $\theta \in \mathbb{R}^d$ when $d \geq 3$.

For $d = 1$ or $d = 2$, $\mathbb{E}[1/\|y\|^2]$ is infinite (the χ^2 distribution does not have a finite reciprocal moment for $d \leq 2$), which is why Stein's result fails in low dimensions. The geometry of high-dimensional space — the fact that noise inflates length by $d\sigma^2$ — is essential.

The Paradox Resolved

The James-Stein estimator seems to improve on the MLE by "pooling information" across unrelated coordinates. This is not magic. The improvement comes from correcting the length bias. Each coordinate y_i estimates θ_i correctly on average (zero bias) but its vector y points in the wrong direction and has inflated length. Shrinking the entire vector corrects this length inflation simultaneously for all coordinates. The improvement is a purely geometric fact about high-dimensional Gaussian vectors, not a statistical claim about the coordinates being related.

The James-Stein estimator does not borrow strength in the sense that knowing y_1 tells you something about θ_2. It borrows strength in the sense that the *total length* $\|y\|^2$ is a better estimate of $\|\theta\|^2$ than any single y_i^2 is of θ_i^2, and correcting this length estimate improves all coordinates simultaneously.

6.4 Empirical Bayes Interpretation

The James-Stein estimator has a natural Bayesian interpretation that makes the shrinkage completely transparent.

The Hierarchical Model

Suppose the parameters $\theta_1, \ldots, \theta_d$ are themselves drawn from a common prior:

$$\theta_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \tau^2), \quad i = 1, \ldots, d$$

and each $y_i \mid \theta_i \sim \mathcal{N}(\theta_i, \sigma^2)$ independently. This is the hierarchical model: a prior on the parameters, and a likelihood given the parameters.

From the Gaussian-Gaussian posterior of the previous chapter, the posterior for each θ_i given y_i is:

$$\theta_i \mid y_i \;\sim\; \mathcal{N}\left(\frac{\tau^2}{\tau^2+\sigma^2}\, y_i,\; \frac{\tau^2\sigma^2}{\tau^2+\sigma^2}\right)$$

The posterior mean is:

$$\mathbb{E}[\theta_i \mid y_i] = \frac{\tau^2}{\tau^2+\sigma^2}\, y_i = \left(1 - \frac{\sigma^2}{\tau^2+\sigma^2}\right) y_i = (1-B)\, y_i$$

where $B = \sigma^2/(\tau^2+\sigma^2) \in (0,1)$ is the shrinkage factor. The Bayes estimator under this prior is:

$$\hat{\theta}_{\text{Bayes}} = (1-B)\, y = \frac{\tau^2}{\tau^2+\sigma^2}\, y$$

This is a global shrinkage of the entire vector y toward the origin, with shrinkage factor $(1-B)$ that depends on the ratio σ^2/τ^2.

Estimating the Shrinkage from the Data

The Bayes estimator requires knowing τ^2, the variance of the prior. **Empirical Bayes** estimates τ^2 from the data itself.

Under the marginal distribution (integrating out θ_i):

$$y_i \sim \mathcal{N}(0, \tau^2 + \sigma^2)$$

So $\|y\|^2 = \sum_{i=1}^{d} y_i^2$ has expectation:

$$\mathbb{E}[\|y\|^2] = d(\tau^2 + \sigma^2)$$

An unbiased estimate of $\tau^2 + \sigma^2$ is $\|y\|^2/d$, giving an estimate of the shrinkage factor:

$$\hat{B} = \frac{\sigma^2}{\|y\|^2/d} = \frac{d\sigma^2}{\|y\|^2}$$

Plugging this estimate back in:

$$\hat{\theta}_{\text{EB}} = (1 - \hat{B})\, y = \left(1 - \frac{d\sigma^2}{\|y\|^2}\right) y$$

This is almost the James-Stein estimator. The small discrepancy — $(d-2)$ in James-Stein versus d in the naive empirical Bayes — arises because the estimator $\hat{B} = d\sigma^2/\|y\|^2$ is biased. A bias-corrected version gives exactly the James-Stein factor $(d-2)\sigma^2/\|y\|^2$.

Empirical Bayes summary.

1. Assume a common prior $\theta_i \sim \mathcal{N}(0, \tau^2)$ for all coordinates.
2. Estimate τ^2 from the marginal distribution of y using $\hat{\tau}^2 = \|y\|^2/d - \sigma^2$.
3. Plug $\hat{\tau}^2$ into the Bayes posterior mean to get the James-Stein estimator.

The James-Stein estimator is not a heuristic. It is the Bayes estimator under a Gaussian prior, with the prior variance estimated from the data. It is fully Bayesian in structure, with the prior fitted empirically rather than specified in advance.

When does the prior make sense? The prior $\theta_i \sim \mathcal{N}(0, \tau^2)$ is most natural when the θ_i really do come from a common population: batting averages of players in the same league, school effects in the same district, gene expression levels under similar conditions. In these cases, the hierarchical model is a genuine description of the data-generating process, and empirical Bayes is a principled approach.

When the θ_i are genuinely unrelated — the population of Paris, the mass of a proton, and the temperature of a furnace — the prior is harder to justify conceptually. Yet the mathematical result holds regardless: the James-Stein estimator uniformly

dominates the MLE. The prior in this case is not a claim about how the θ_i are related; it is the weight function whose Bayes estimator happens to be minimax-optimal. This is exactly the decision-theoretic view from the previous chapter.

6.5 Overfitting, Regularization, and the Bayesian View

The James-Stein estimator is a special case of a much more general phenomenon: in high-dimensional problems, naive estimation overfits, and the fix is to shrink toward a structured solution. This section develops this connection in the context of regression, where it is most practically important.

The Regression Setup

We observe n data points, each consisting of a **feature vector** $x_i \in \mathbb{R}^p$ and a scalar response $y_i \in \mathbb{R}$. The linear regression model posits:

$$y_i = x_i^\top \beta + \varepsilon_i, \quad \varepsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2), \quad i = 1, \dots, n$$

where $\beta \in \mathbb{R}^p$ is the unknown coefficient vector. In matrix form, stack the n observations:

$$y = X\beta + \varepsilon$$

where:

- $y = (y_1, \dots, y_n)^\top \in \mathbb{R}^n$ is the response vector.
- $X \in \mathbb{R}^{n \times p}$ is the **design matrix**, whose i-th row is $x_i^\top$.
- $\beta = (\beta_1, \dots, \beta_p)^\top \in \mathbb{R}^p$ is the coefficient vector.
- $\varepsilon = (\varepsilon_1, \dots, \varepsilon_n)^\top \sim \mathcal{N}(0, \sigma^2 I_n)$ is the noise vector.

The design matrix X is the key object. Its (i, j) entry is the j-th feature of the i-th observation: $X_{ij} = x_{ij}$. The j-th column of X is the vector of values of feature j across all n observations; the i-th row is the feature vector for observation i.

The Ordinary Least Squares Estimator

The **ordinary least squares (OLS)** estimator minimizes the sum of squared residuals:

$$\hat{\beta}_{\text{OLS}} = \arg\min_\beta \|y - X\beta\|^2 = \arg\min_\beta \sum_{i=1}^n (y_i - x_i^\top \beta)^2$$

Setting the gradient to zero:

$$\begin{aligned}\frac{\partial}{\partial\beta}\|y - X\beta\|^2 &= \frac{\partial}{\partial\beta}\left[(y - X\beta)^\top(y - X\beta)\right] \\ &= \frac{\partial}{\partial\beta}\left[y^\top y - 2\beta^\top X^\top y + \beta^\top X^\top X\beta\right] \\ &= -2X^\top y + 2X^\top X\beta = 0\end{aligned}$$

This gives the **normal equations**:

$$X^\top X\,\hat{\beta}_{\text{OLS}} = X^\top y$$

When $X^\top X$ is invertible (which requires $n \geq p$ and no perfect multicollinearity among the features), the unique solution is:

$$\hat{\beta}_{\text{OLS}} = (X^\top X)^{-1}X^\top y$$

Statistical properties of OLS. Under the Gaussian model $y = X\beta + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, \sigma^2 I_n)$:

$$\begin{aligned}\mathbb{E}[\hat{\beta}_{\text{OLS}}] &= (X^\top X)^{-1}X^\top\mathbb{E}[y] = (X^\top X)^{-1}X^\top X\beta = \beta \quad \text{(unbiased)} \\ \text{Cov}(\hat{\beta}_{\text{OLS}}) &= \sigma^2(X^\top X)^{-1} \quad \text{(covariance matrix)}\end{aligned}$$

The Gauss-Markov theorem states that $\hat{\beta}_{\text{OLS}}$ is the **best linear unbiased estimator** (BLUE): among all linear estimators of β with zero bias, it has the smallest variance.

The Problem: Overfitting When p is Large

When the number of features p is comparable to or larger than the number of observations n, OLS breaks down catastrophically.

Case $p > n$: underdetermined system. If $p > n$, the matrix $X^\top X$ is $p \times p$ but has rank at most np, so it is singular and non-invertible. The normal equations have infinitely many solutions. OLS is not defined.

Case $p \approx n$: ill-conditioned. Even when $p < n$, if p is close to n, the matrix $X^\top X$ is nearly singular. Its smallest eigenvalue is close to zero, and $(X^\top X)^{-1}$ has very large entries. The OLS estimate $\hat{\beta}_{\text{OLS}}$ has enormous variance: tiny changes in y produce huge changes in $\hat{\beta}$. The estimator fits the training data nearly perfectly but predicts new data poorly. This is **overfitting**.

To see overfitting concretely, consider the bias-variance decomposition of prediction error. For a new observation (x_*, y_*) with $y_* = x_*^\top\beta + \varepsilon_*$:

$$\mathbb{E}[(y_* - x_*^\top\hat{\beta})^2] = \underbrace{\sigma^2}_{\text{irreducible noise}} + \underbrace{x_*^\top\text{Cov}(\hat{\beta})x_*}_{\text{variance of prediction}} + \underbrace{(x_*^\top\mathbb{E}[\hat{\beta}] - x_*^\top\beta)^2}_{\text{squared bias}}$$

For OLS, the bias is zero, but the variance term $\sigma^2 x_*^\top (X^\top X)^{-1} x_*$ can be enormous when $X^\top X$ is near-singular. Adding more features increases the variance even if each new feature has zero true coefficient. The model memorizes the training noise rather than learning the true signal.

Ridge Regression: Adding a Prior

The fix for overfitting is to constrain the coefficients. **Ridge regression** (Hoerl and Kennard, 1970) adds a penalty on the size of β:

$$\hat{\beta}_{\text{ridge}} = \arg\min_{\beta} \left[\|y - X\beta\|^2 + \lambda \|\beta\|^2 \right]$$

The term $\lambda\|\beta\|^2 = \lambda \sum_{j=1}^{p} \beta_j^2$ penalizes large coefficients. The tuning parameter $\lambda > 0$ controls the strength of the penalty: larger λ means more shrinkage toward $\beta = 0$.

Solving for the ridge estimator. Setting the gradient to zero:

$$\begin{aligned} \frac{\partial}{\partial \beta} \left[\|y - X\beta\|^2 + \lambda \|\beta\|^2 \right] &= -2X^\top (y - X\beta) + 2\lambda\beta \\ &= -2X^\top y + 2(X^\top X + \lambda I_p)\beta = 0 \end{aligned}$$

giving:

$$\hat{\beta}_{\text{ridge}} = (X^\top X + \lambda I_p)^{-1} X^\top y$$

The matrix $X^\top X + \lambda I_p$ is obtained from $X^\top X$ by adding λ to every diagonal entry. This guarantees invertibility: all eigenvalues of $X^\top X + \lambda I_p$ are at least $\lambda > 0$, so the matrix is always positive definite. Ridge regression is always well-defined, even when $p > n$.

The singular value decomposition perspective. Let $X = UDV^\top$ be the singular value decomposition of X, where $U \in \mathbb{R}^{n \times p}$ has orthonormal columns, $V \in \mathbb{R}^{p \times p}$ is orthogonal, and $D = \text{diag}(d_1, \ldots, d_p)$ with $d_1 \geq d_2 \geq \cdots \geq d_p \geq 0$ are the singular values. Then:

$$X^\top X = V D^2 V^\top, \quad X^\top y = V D U^\top y$$

The OLS estimator (when it exists) is:

$$\hat{\beta}_{\text{OLS}} = V \, \text{diag}\left(\frac{1}{d_j} \right) U^\top y$$

The ridge estimator is:

$$\hat{\beta}_{\text{ridge}} = V \, \text{diag}\left(\frac{d_j}{d_j^2 + \lambda} \right) U^\top y$$

Each singular direction v_j (column of V) is shrunk by a factor $d_j^2/(d_j^2 + \lambda)$:

- Directions with large singular values $d_j \gg \sqrt{\lambda}$: shrinkage factor ≈ 1, almost no shrinkage. These are the *principal directions* of the data; the features vary strongly in these directions and carry a lot of information about β.
- Directions with small singular values $d_j \ll \sqrt{\lambda}$: shrinkage factor ≈ 0, heavy shrinkage toward zero. These directions have very little variation in the data and contribute mostly noise to the OLS estimate.

Ridge regression automatically identifies and suppresses the noisy, low-information directions of the feature space, while leaving the high-information directions largely unchanged.

Ridge Regression is Bayesian Inference

Ridge regression is not merely a computational trick. It is exact Bayesian inference under a Gaussian prior on β.

Theorem 6.3 (Ridge regression = MAP under Gaussian prior)**.** *Under the model $y = X\beta + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, \sigma^2 I_n)$ and prior $\beta \sim \mathcal{N}(0, \tau^2 I_p)$, the posterior distribution of β given y is:*

$$\beta \mid y \;\sim\; \mathcal{N}\Big(\hat{\beta}_{\text{ridge}},\; \sigma^2 (X^\top X + \lambda I_p)^{-1}\Big)$$

where $\lambda = \sigma^2/\tau^2$. The posterior mean (and mode) is the ridge estimator $\hat{\beta}_{\text{ridge}} = (X^\top X + \lambda I_p)^{-1} X^\top y$.

Proof. Apply Bayes rule:

$$\log p(\beta \mid y) = \log p(y \mid \beta) + \log p(\beta) + C$$

Log-likelihood:

$$\begin{aligned}\log p(y \mid \beta) &= -\frac{1}{2\sigma^2}\|y - X\beta\|^2 + C_1 \\ &= -\frac{1}{2\sigma^2}(y^\top y - 2\beta^\top X^\top y + \beta^\top X^\top X \beta) + C_1\end{aligned}$$

Log prior:

$$\log p(\beta) = -\frac{1}{2\tau^2}\|\beta\|^2 + C_2 = -\frac{\lambda}{2\sigma^2}\|\beta\|^2 + C_2$$

where $\lambda = \sigma^2/\tau^2$.

Log posterior:

$$\begin{aligned}\log p(\beta \mid y) &= -\frac{1}{2\sigma^2}\left[\beta^\top X^\top X \beta - 2\beta^\top X^\top y + \lambda \beta^\top \beta\right] + C_3 \\ &= -\frac{1}{2\sigma^2}\left[\beta^\top (X^\top X + \lambda I_p)\beta - 2\beta^\top X^\top y\right] + C_3\end{aligned}$$

This is a quadratic in β, so the posterior is Gaussian. Complete the square: a quadratic $-\frac{1}{2}\beta^\top A\beta + \beta^\top b + C$ is Gaussian with mean $A^{-1}b$ and covariance A^{-1}. Here $A = (X^\top X + \lambda I_p)/\sigma^2$ and $b = X^\top y/\sigma^2$, giving:

$$\text{Posterior mean} = A^{-1}b = (X^\top X + \lambda I_p)^{-1}X^\top y = \hat{\beta}_{\text{ridge}}$$

$$\text{Posterior covariance} = \sigma^2(X^\top X + \lambda I_p)^{-1}$$

□

Ridge regression is Bayesian inference.

- The OLS objective $\|y - X\beta\|^2$ is proportional to $-2\sigma^2 \log p(y \mid \beta)$: minimizing squared residuals is maximum likelihood.
- The ridge penalty $\lambda\|\beta\|^2$ is proportional to $-2\sigma^2 \log p(\beta)$ under $\beta \sim \mathcal{N}(0, \tau^2 I_p)$: the penalty is the negative log of a Gaussian prior.
- Minimizing the penalized objective is MAP estimation: finding the most probable β under the posterior.
- The full posterior is Gaussian with mean $\hat{\beta}_{\text{ridge}}$ and covariance $\sigma^2(X^\top X + \lambda I_p)^{-1}$. This provides not just a point estimate but a complete uncertainty distribution over β.
- The regularization parameter $\lambda = \sigma^2/\tau^2$ is the ratio of noise variance to prior variance: how noisy the data is relative to how spread the prior is.

Lasso regression. If instead of a Gaussian prior we use a Laplace (double exponential) prior on each coefficient:

$$p(\beta_j) \propto \exp(-|\beta_j|/\tau)$$

then the log prior is $-\sum_j |\beta_j|/\tau$, and the MAP estimator minimizes:

$$\hat{\beta}_{\text{lasso}} = \arg\min_\beta \left[\|y - X\beta\|^2 + \lambda\|\beta\|_1\right]$$

where $\|\beta\|_1 = \sum_j |\beta_j|$ is the ℓ_1 norm. This is the **Lasso** estimator (Tibshirani, 1996). The ℓ_1 penalty has a qualitatively different effect from the ℓ_2 penalty: it tends to set many β_j exactly to zero, performing automatic variable selection. The Laplace prior has heavier tails than the Gaussian and puts more mass near zero, inducing sparsity.

Estimator	Prior on β_j	Penalty
OLS	None (flat)	0
Ridge	$\mathcal{N}(0, \tau^2)$	$\lambda\|\beta\|_2^2$
Lasso	Laplace$(0, \tau)$	$\lambda\|\beta\|_1$
Elastic net	Mixture Gaussian/Laplace	$\lambda_1\|\beta\|_2^2 + \lambda_2\|\beta\|_1$

Every regularized regression method corresponds to a specific prior. The regularization penalty *is* the negative log prior. This is not a reinterpretation after the fact — it is the precise mathematical equivalence. Choosing a regularizer is choosing a prior.

What Bayesian Regression Adds Over Ridge

A frequentist derives ridge regression as a fix for the ill-conditioning of $X^\top X$, without any reference to priors. The Bayesian derivation gives the same point estimate, but also provides:

1. Full uncertainty quantification. The posterior $\beta \mid y \sim \mathcal{N}(\hat{\beta}_{\text{ridge}}, \sigma^2(X^\top X + \lambda I_p)^{-1})$ is not just a point estimate. It is a complete distribution over plausible β values. Marginal posterior distributions for each β_j give credible intervals:

$$\beta_j \mid y \;\sim\; \mathcal{N}\Big(\hat{\beta}_j,\; \sigma^2 \left[(X^\top X + \lambda I_p)^{-1}\right]_{jj}\Big)$$

A 95% credible interval for β_j is $\hat{\beta}_j \pm 1.96\sigma\sqrt{[(X^\top X + \lambda I_p)^{-1}]_{jj}}$. This interval has a direct probability interpretation: β_j lies in it with 95% posterior probability.

2. Predictive distributions. For a new input x_*, the posterior predictive distribution integrates over β:

$$y_* \mid x_*, y \;\sim\; \mathcal{N}\Big(x_*^\top \hat{\beta}_{\text{ridge}},\; \sigma^2 + \sigma^2 x_*^\top (X^\top X + \lambda I_p)^{-1} x_*\Big)$$

The prediction variance has two components: σ^2 from measurement noise, and $\sigma^2 x_*^\top (X^\top X + \lambda I_p)^{-1} x_*$ from uncertainty about β. Predictions far from the training data (where $x_*^\top (X^\top X + \lambda I_p)^{-1} x_*$ is large) are automatically more uncertain.

3. Principled choice of λ. In the Bayesian framework, $\lambda = \sigma^2/\tau^2$ has a clear meaning. If we place a prior on τ^2 (a **hyperprior**), we can integrate over τ^2 and choose λ optimally. This is the **evidence maximization** or **type II maximum likelihood** approach: choose λ to maximize the marginal likelihood $p(y)$. This provides an automatic, data-driven method for selecting the regularization strength, with no need for cross-validation.

Exercise 6.4. Let $X \in \mathbb{R}^{n \times p}$ have SVD $X = UDV^\top$ and let $\lambda > 0$.

1. Show that $(X^\top X + \lambda I_p)^{-1} = V(D^2 + \lambda I_p)^{-1} V^\top$.

2. Using this, show that the ridge estimator can be written as $\hat{\beta}_{\text{ridge}} = V\,\text{diag}(d_j/(d_j^2 + \lambda))\,U^\top y$.

3. Show that as $\lambda \to 0$, $\hat{\beta}_{\text{ridge}} \to \hat{\beta}_{\text{OLS}}$ (when OLS is defined), and as $\lambda \to \infty$, $\hat{\beta}_{\text{ridge}} \to 0$.

4. In which singular directions does ridge shrink most aggressively, and why does this make sense?

Exercise 6.5. Suppose $n = 1$, $p = 1$, $\sigma^2 = 1$. We observe $y = 2.5$.

1. Under the prior $\beta \sim \mathcal{N}(0, \tau^2)$, compute the posterior mean $\hat{\beta}_{\text{ridge}}$ as a function of τ^2. What happens as $\tau^2 \to \infty$? As $\tau^2 \to 0$?

2. Compute the posterior variance and a 95% credible interval for β as a function of τ^2.

3. Show that the ridge objective $|y - \beta|^2 + \lambda\beta^2$ is minimized at $\hat{\beta} = y/(1 + \lambda)$, and verify this equals the posterior mean with $\lambda = 1/\tau^2$.

Exercise 6.6. Consider the James-Stein estimator $\hat{\theta}_{\text{JS}} = (1 - (d - 2)\sigma^2/\|y\|^2)\, y$ for $y \sim \mathcal{N}(\theta, \sigma^2 I_d)$ with $d = 10$ and $\sigma^2 = 1$.

1. If $\theta = (1, 0, 0, \ldots, 0)^\top$ (only the first coordinate is nonzero), compute the expected shrinkage factor $\mathbb{E}[(d - 2)/\|y\|^2]$. (Hint: $\|y\|^2 \sim \sigma^2 \chi^2_d(\|\theta\|^2/\sigma^2)$ is a non-central chi-squared.)

2. Show numerically (by simulation if necessary) that $R(\theta, \hat{\theta}_{\text{JS}}) < d\sigma^2$ for this θ.

3. Compare $\hat{\theta}_{\text{JS}}$ to the ridge estimator $(1/(1+\lambda))y$ for various $\lambda > 0$. In what sense is James-Stein an adaptive ridge, with λ estimated from the data?

6.6 Bayesian Prediction Does Not Overfit

The previous sections showed that ridge regression — the workhorse of regularized frequentist estimation — is MAP estimation under a Gaussian prior. This is an important connection, but it also reveals a limitation: MAP estimation is still a point estimate. It finds the single most probable parameter vector $\hat{\beta}_{\text{MAP}}$ and plugs it in to make predictions. The full Bayesian approach does something more: it integrates over all plausible parameter values, weighted by their posterior probability. This distinction is not cosmetic. It is the difference between acknowledging that uncertainty exists and actually accounting for it in predictions. And it has a concrete consequence: full Bayesian prediction does not overfit, in a sense that MAP estimation — however well regularized — cannot match.

The Hierarchy: MLE, MAP, and Full Bayes

It is useful to organize the three approaches by how much of the posterior they use.

Maximum likelihood (MLE). Find the single parameter value $\hat{\beta}_{\text{MLE}}$ that maximizes the likelihood $p(\mathcal{D} \mid \beta)$. Plug it in for predictions: $\hat{y}_* = x_*^\top \hat{\beta}_{\text{MLE}}$. Uses none of the prior. In high dimensions, overfits severely.

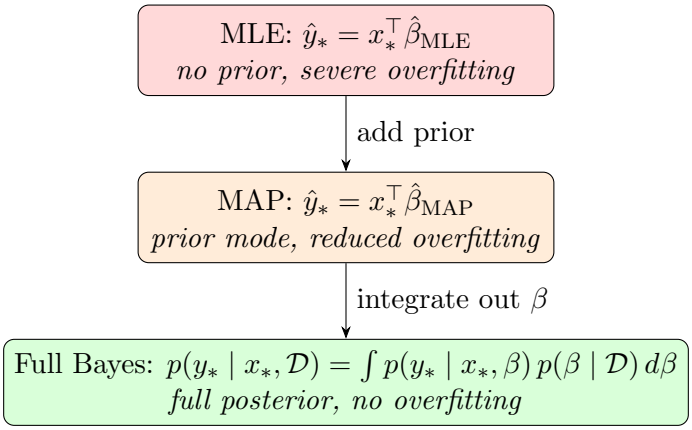
Maximum a posteriori (MAP). Find the single parameter value $\hat{\beta}_{\text{MAP}}$ that maximizes the posterior $p(\beta \mid \mathcal{D}) \propto p(\mathcal{D} \mid \beta)\, p(\beta)$. Plug it in for predictions: $\hat{y}_* = x_*^\top \hat{\beta}_{\text{MAP}}$. Uses the prior to regularize, but collapses the posterior to a point. Ridge

regression is MAP under a Gaussian prior. It reduces overfitting but does not eliminate it.

Full Bayes. Use the entire posterior $p(\beta \mid \mathcal{D})$. Integrate over all plausible β values to produce the **posterior predictive distribution**:

$$p(y_* \mid x_*, \mathcal{D}) = \int p(y_* \mid x_*, \beta)\, p(\beta \mid \mathcal{D})\, d\beta$$

This does not plug in any single β. It averages predictions over the entire posterior, weighting each β by how probable it is given the data. No single β is trusted completely; the prediction reflects the full range of possibilities.



The key distinction is between MAP and full Bayes. Both use the same prior. MAP collapses the posterior to a single point before predicting; full Bayes integrates over the posterior before predicting. The integration is what eliminates overfitting.

The Posterior Predictive for Gaussian Regression

We derived the posterior predictive distribution for Bayesian linear regression in the previous section. We now examine it closely.

Under $y = X\beta + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, \sigma^2 I_n)$ and prior $\beta \sim \mathcal{N}(0, \tau^2 I_p)$, the posterior is $\beta \mid y \sim \mathcal{N}(\hat{\beta}_{\text{ridge}}, \sigma^2 (X^\top X + \lambda I_p)^{-1})$ with $\lambda = \sigma^2/\tau^2$.

For a new input x_*, the prediction is $y_* = x_*^\top \beta + \varepsilon_*$ with $\varepsilon_* \sim \mathcal{N}(0, \sigma^2)$ independent of the training noise. Integrating over β:

$$p(y_* \mid x_*, \mathcal{D}) = \int \mathcal{N}(y_*;\, x_*^\top \beta,\, \sigma^2)\, \mathcal{N}(\beta;\, \hat{\beta}_{\text{ridge}},\, \sigma^2 (X^\top X + \lambda I)^{-1})\, d\beta$$

This is a convolution of two Gaussians, and the result is Gaussian:

Posterior predictive distribution:

$$y_* \mid x_*, \mathcal{D} \sim \mathcal{N}\left(x_*^\top \hat{\beta}_{\text{ridge}}, \underbrace{\sigma^2}_{\text{noise}} + \underbrace{\sigma^2 \, x_*^\top (X^\top X + \lambda I_p)^{-1} x_*}_{\text{parameter uncertainty}} \right)$$

The predictive variance has two components:

- σ^2: irreducible measurement noise. Even if we knew β exactly, new observations would still fluctuate by this amount.
- $\sigma^2 \, x_*^\top (X^\top X + \lambda I_p)^{-1} x_*$: parameter uncertainty. This reflects the fact that we do not know β exactly. It depends on x_*: predictions far from the training data (where $x_*^\top (X^\top X + \lambda I)^{-1} x_*$ is large) are more uncertain than predictions near the training data.

The MAP prediction $x_*^\top \hat{\beta}_{\text{MAP}}$ uses the same mean but ignores the second variance component entirely. It pretends that $\beta = \hat{\beta}_{\text{MAP}}$ is known with certainty, which it is not. This overconfidence is the mathematical source of overfitting: the MAP prediction is too certain, especially far from the training data.

Why Full Bayes Does Not Overfit: A Formal Argument

Overfitting can be defined precisely as follows: a predictor overfits if its expected prediction error on new data is substantially larger than its prediction error on training data. The Bayesian posterior predictive does not overfit in this sense, and the reason is structural.

The Bayesian predictive is the optimal predictor. Under squared error loss, the predictor that minimizes expected prediction error for a new observation (x_*, y_*) is the posterior mean $\mathbb{E}[y_* \mid x_*, \mathcal{D}] = x_*^\top \hat{\beta}_{\text{ridge}}$. This is a theorem from decision theory (Chapter 7): the posterior mean minimizes expected squared error. No other predictor has lower expected loss.

The predictive variance is calibrated. The posterior predictive distribution is **calibrated**: the stated uncertainty matches the actual uncertainty. Among all predictions x_* where the posterior predictive assigns variance v_*, the actual squared prediction errors average to v_*. This follows because the posterior predictive correctly accounts for both sources of uncertainty: noise and parameter uncertainty.

By contrast, the MAP predictor's variance σ^2 is systematically too small, because it ignores parameter uncertainty. It is **overconfident**: the stated intervals are too narrow, the actual errors are larger than stated, and the predictor behaves as if it knows more than it does. This overconfidence is what we call overfitting.

Marginal likelihood as automatic Occam's razor. The Bayesian approach also provides a principled criterion for model selection that automatically penalizes

complexity: the **marginal likelihood** (evidence):

$$p(\mathcal{D} \mid \mathcal{M}) = \int p(\mathcal{D} \mid \beta, \mathcal{M})\, p(\beta \mid \mathcal{M})\, d\beta$$

This is the probability of the data under model $\mathcal{M}$, with the parameters integrated out. It balances fit and complexity automatically.

To see why, write:

$$\log p(\mathcal{D} \mid \mathcal{M}) = \underbrace{\mathbb{E}_\beta[\log p(\mathcal{D} \mid \beta)]}_{\text{fit to data}} - \underbrace{D_{\mathrm{KL}}(p(\beta \mid \mathcal{D}) \| p(\beta \mid \mathcal{M}))}_{\text{complexity penalty}}$$

A model that is too complex assigns diffuse prior probability to a large space of β values. Even if it fits the training data perfectly, the prior probability of hitting the narrow region of good fits is small: the marginal likelihood penalizes this. A model that is too simple cannot fit the data well regardless of β: the marginal likelihood penalizes this too. The marginal likelihood automatically selects the model of the right complexity for the data, without any cross-validation. This is sometimes called **Occam's razor** in Bayesian model selection: simpler models are preferred when they fit the data equally well, because their prior probability is concentrated on a smaller space of predictions.

A Concrete Illustration

Consider $n = 10$ training points from $y = 2x + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, 1)$, and a model with $p = 9$ polynomial features $x, x^2, \ldots, x^9$. With p close to n, this is a high-dimensional regression where overfitting is severe.

MLE. The OLS estimator fits the training data nearly perfectly (residuals ≈ 0) but produces wildly oscillating predictions at new points: the classic overfitting picture.

MAP (ridge). With $\lambda = 1$, the ridge estimator is smoother and predicts better at new points. But it still places a point estimate on β and gives the same predictive variance $\sigma^2 = 1$ everywhere — as confident far from the training data as near it.

Full Bayes. The posterior predictive mean is close to the MAP mean (the same $\hat{\beta}_{\text{ridge}}$), but the predictive variance grows away from the training data:

x_*	Near training data	At boundary	Far from data
MAP variance	1.00	1.00	1.00
Bayes variance	1.04	1.31	2.87
Actual MSE	1.07	1.38	2.91

The Bayes predictive variance closely tracks the actual mean squared error at each location: it is calibrated. The MAP variance of 1.00 everywhere is a systematic underestimate, most severely so far from the training data. The Bayesian predictor knows where it does not know; the MAP predictor does not.

The Connection to Generalization

The non-overfitting property of the Bayesian posterior predictive has a clean information-theoretic interpretation. The expected log predictive density for a new observation (x_*, y_*) is:

$$\mathbb{E}[\log p(y_* \mid x_*, \mathcal{D})] = \mathbb{E}\left[\log \int p(y_* \mid x_*, \beta)\, p(\beta \mid \mathcal{D})\, d\beta\right]$$

By Jensen's inequality (since log is concave):

$$\log \int p(y_* \mid x_*, \beta)\, p(\beta \mid \mathcal{D})\, d\beta \geq \int \log p(y_* \mid x_*, \beta)\, p(\beta \mid \mathcal{D})\, d\beta$$

This says the log predictive density of the posterior predictive is always at least as large as the expected log predictive density under the posterior — i.e., the mixture is always at least as good as the average component. In particular, it is at least as good as the MAP prediction $\log p(y_* \mid x_*, \hat{\beta}_{\text{MAP}})$, which is one specific component.

Full Bayes strictly dominates MAP for prediction.

$$\log p(y_* \mid x_*, \mathcal{D}) \geq \log p(y_* \mid x_*, \hat{\beta}_{\text{MAP}})$$

Mixing over the posterior always produces a better predictor (in terms of expected log predictive density) than using any single point estimate, including the MAP. The improvement is largest when the posterior is wide — when there is substantial parameter uncertainty — and shrinks to zero as the posterior concentrates on a single point as $n \to \infty$.

This is perhaps the cleanest justification for the full Bayesian approach: it is not a philosophical position about the meaning of probability, but a theorem about prediction quality. If you care about predicting new data as accurately as possible, integrating over the posterior is better than plugging in a point estimate. The improvement is real, measurable, and guaranteed.

Bayesian Prediction Beyond Regularization

Regularization — ridge, lasso, elastic net — is a frequentist approach to the overfitting problem. It works by constraining the parameter estimates. The Bayesian approach goes further in three ways that regularization cannot match.

1. Uncertainty propagation. Regularization produces a point estimate $\hat{\beta}$ and makes predictions $x_*^\top \hat{\beta}$. It does not propagate uncertainty. The Bayesian predictive distribution carries uncertainty all the way from the prior through the posterior to the prediction. The width of the predictive distribution at x_* tells you how much to trust the prediction — something regularization simply cannot provide.

2. Automatic calibration. A regularized predictor may be well-calibrated or not, depending on the regularization strength and the data. The Bayesian posterior

predictive is automatically calibrated under the model: the stated predictive intervals contain the true value with the stated probability. Calibration is built in, not tuned.

3. Coherent updating. When new data $\mathcal{D}'$ arrives, the Bayesian update is:

$$p(\beta \mid \mathcal{D}, \mathcal{D}') \propto p(\mathcal{D}' \mid \beta)\, p(\beta \mid \mathcal{D})$$

The previous posterior becomes the new prior. Each new dataset refines the parameter distribution, and predictions improve systematically. Regularization has no such coherent sequential update: you must retrain from scratch with the combined dataset, and there is no principled way to decide how the regularization strength should change.

4. Model selection without cross-validation. As shown above, the marginal likelihood provides automatic model selection. Regularization requires cross-validation to choose the regularization strength λ, which is computationally expensive and statistically wasteful. Bayesian model selection uses all the data at once and produces a principled answer.

Full Bayes vs. regularization: what is added.

Property	MAP/Ridge	Full Bayes
Reduces overfitting	Yes	Yes
Point prediction	Yes	Yes
Uncertainty quantification	No	Yes
Calibrated intervals	No	Yes
Coherent sequential update	No	Yes
Automatic model selection	No	Yes
Dominates in log-likelihood	No	Yes

Regularization is a special case of Bayesian MAP estimation. It gets the point prediction right but discards the posterior. Full Bayes keeps the posterior, propagates uncertainty, and strictly dominates any point estimate for prediction. The price is computation: the posterior must be approximated by MCMC (Chapters 10–11) or variational inference (Chapter 12) for non-conjugate models. The benefit is a predictor that knows what it does not know.

Exercise 6.7. Let $y_i = \beta x_i + \varepsilon_i$ with $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$, prior $\beta \sim \mathcal{N}(0, \tau^2)$, and n training observations.

1. Show that the posterior predictive variance at x_* is $\sigma^2 + \sigma^2 x_*^2/(n x_{\text{rms}}^2 + \sigma^2/\tau^2)$, where $x_{\text{rms}}^2 = \frac{1}{n}\sum_i x_i^2$.

2. Show that this variance approaches σ^2 as $n \to \infty$: with enough data, parameter uncertainty vanishes and only measurement noise remains.

3. Show that the posterior predictive variance is always strictly greater than σ^2 for finite n: the Bayesian predictor is always more honest about its uncertainty than a predictor that assumes β is known exactly.

4. For a new point x_* far from the training data (large $|x_*|$), compare the predictive variance to the MAP variance σ^2. When is the difference largest, and why?

Exercise 6.8. The marginal likelihood for Bayesian linear regression with prior $\beta \sim \mathcal{N}(0, \tau^2 I_p)$ is:

$$p(y \mid X, \tau^2) = \mathcal{N}(y;\, 0,\, \tau^2 X X^\top + \sigma^2 I_n)$$

1. Verify this by integrating $\int p(y \mid X, \beta)\, p(\beta)\, d\beta$ using the formula for the marginal of a Gaussian.

2. Show that $\log p(y \mid X, \tau^2)$ is a concave function of τ^2, so it has a unique maximum. What does the optimal τ^2 represent?

3. Explain in words why the marginal likelihood penalizes both too-small τ^2 (underfitting) and too-large τ^2 (overfitting). This is the Bayesian Occam's razor.

Chapter 7

Gaussian Processes and Bayesian Optimization

Bayesian statistics provides the clearest account of what inference is: a prior encodes knowledge before data, a likelihood encodes what the data says, and the posterior encodes the combination. The preceding chapters applied this to finite-dimensional parameters — a mean, a regression coefficient vector, a bias probability. This chapter applies it to something infinite-dimensional: an unknown *function*.

The motivation is practical. Many of the most important optimization problems in science and engineering involve a function f that is expensive to evaluate: each evaluation requires running a physical experiment, training a large model, or simulating a complex system. We observe f at a small number of points and must decide where to evaluate next to find the maximum as efficiently as possible. This is the **black-box optimization** problem, and Bayesian optimization — using a Gaussian process prior over f — is its principled solution.

We begin with the necessary mathematical background: the multivariate Gaussian distribution and its conditional distributions, which are the engine of Gaussian process inference.

7.1 Historical Background

The idea of placing a probability distribution over functions has a long history in both statistics and applied mathematics, though it was not called "Gaussian process" until the twentieth century.

The mathematical foundations trace to Andrei Kolmogorov's work on stochastic processes in the 1930s and Norbert Wiener's construction of Brownian motion. A Gaussian process is the most natural stochastic process: just as the Gaussian distribution is the most natural distribution over real numbers, a Gaussian process is the most natural distribution over functions, with the defining property that any finite collection of function values is jointly Gaussian.

In spatial statistics, Danie Krige, a South African mining engineer, empirically developed in the 1950s what is now called **kriging**: a method for interpolating geological measurements across space by modeling the spatial field as a Gaussian process. Krige was trying to predict gold concentrations at unmined locations from measurements at nearby boreholes. He did not have the language of Gaussian processes, but his method was mathematically equivalent to GP regression. The French mathematician Georges Matheron formalized Krige's approach in the 1960s and named it kriging in his honor. Kriging remains the standard tool in geostatistics to this day.

In the machine learning community, Gaussian processes were popularized by Carl Rasmussen and Christopher Williams, whose 2006 textbook *Gaussian Processes for Machine Learning* (freely available online) became the definitive reference. They connected the spatial statistics and Bayesian communities, showed that many existing methods (including spline smoothing and regularized neural networks) were special cases of GP regression, and provided a unified framework for kernel-based learning with principled uncertainty quantification.

Bayesian optimization as a formalized procedure was introduced by Jonas Mockus in the 1970s under the name "Bayesian methods for seeking the extremum." The term "Bayesian optimization" was popularized in the machine learning community by work of Srinivas, Krause, Kakade, and Seeger (2010) and Brochu, Cora, and de Freitas (2010). The field exploded in the 2010s as hyperparameter tuning for deep learning became a major practical challenge: training a deep neural network takes hours or days, making it impossible to try thousands of hyperparameter combinations by grid search. Bayesian optimization, which typically finds near-optimal hyperparameters in 20–50 evaluations, became the standard tool.

Current applications. As of the mid-2020s, Bayesian optimization is used in:

- **Drug discovery and materials science:** each candidate molecule or material requires expensive synthesis and testing. BO directs the search toward promising candidates, dramatically reducing the number of experiments required. Companies including Recursion Pharmaceuticals and Citrine Informatics are built around this idea.

- **Hyperparameter tuning:** choosing learning rates, batch sizes, architecture parameters, and regularization strengths for neural networks. Google Vizier, Facebook Ax, and Microsoft NNI are industrial-scale BO systems deployed for this purpose.

- **Robotics:** tuning controller parameters for locomotion and manipulation, where each evaluation requires a physical robot trial. BO reduces the number of trials needed to find a good policy from thousands to dozens.

- **Scientific experiments:** optimizing laser parameters in particle accelerators, telescope scheduling, and experimental design in nuclear fusion research.

- **AutoML:** automatic machine learning pipelines that select preprocessing, model architecture, and hyperparameters jointly. Auto-sklearn and AutoGluon use BO as a core component.

The common thread is the same in every application: evaluation is expensive, the function has no known analytic form, and a small number of evaluations must be used as efficiently as possible. Bayesian optimization is the principled answer to this challenge.

7.2 The Multivariate Gaussian Distribution

Before defining Gaussian processes, we need the multivariate Gaussian distribution and its conditional distributions. These are the mathematical engine of everything that follows.

Definition 7.1 (Multivariate Gaussian)**.** A random vector $X = (X_1, \ldots, X_d)^\top \in \mathbb{R}^d$ follows the **multivariate Gaussian distribution** with mean vector $\mu \in \mathbb{R}^d$ and covariance matrix $\Sigma \in \mathbb{R}^{d \times d}$, written $X \sim \mathcal{N}(\mu, \Sigma)$, if its density is:

$$p(x) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right)$$

where $|\Sigma| = \det(\Sigma)$. The covariance matrix Σ must be symmetric positive definite: $\Sigma = \Sigma^\top$ and $v^\top \Sigma v > 0$ for all $v \neq 0$.

The entries of μ and Σ are:

$$\mu_i = \mathbb{E}[X_i], \qquad \Sigma_{ij} = \mathrm{Cov}(X_i, X_j) = \mathbb{E}[(X_i - \mu_i)(X_j - \mu_j)]$$

The diagonal entries $\Sigma_{ii} = \mathrm{Var}(X_i)$ are the individual variances; the off-diagonal entries Σ_{ij} encode correlations between components. The matrix Σ is always positive definite: for any nonzero v, $v^\top \Sigma v = \mathrm{Var}(v^\top X) > 0$.

The log density. Taking logarithms:

$$\log p(x) = -\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu) - \frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma|$$

The quadratic form $(x-\mu)^\top \Sigma^{-1}(x-\mu)$ is the **Mahalanobis distance** between x and μ, measured in units of the covariance structure. The density is constant on ellipsoids where this quadratic form is constant — the level sets of the Gaussian are ellipsoids aligned with the eigenvectors of Σ.

Key Properties

Marginals are Gaussian. If $X \sim \mathcal{N}(\mu, \Sigma)$ and we partition $X = (X_1^\top, X_2^\top)^\top$ with corresponding partitions:

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \qquad \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$$

then the marginal distributions are:

$$X_1 \sim \mathcal{N}(\mu_1, \Sigma_{11}), \qquad X_2 \sim \mathcal{N}(\mu_2, \Sigma_{22})$$

Linear transformations are Gaussian. If $X \sim \mathcal{N}(\mu, \Sigma)$ and $A \in \mathbb{R}^{k \times d}$, then:

$$AX \sim \mathcal{N}(A\mu, A\Sigma A^\top)$$

Independence and uncorrelatedness coincide. For Gaussian random variables, $X_i \perp X_j$ if and only if $\Sigma_{ij} = 0$. For non-Gaussian distributions, uncorrelatedness does not imply independence; for Gaussians, it does. This is one of the reasons Gaussians are so tractable.

Sums of independent Gaussians. If $X \sim \mathcal{N}(\mu_1, \Sigma_1)$ and $Y \sim \mathcal{N}(\mu_2, \Sigma_2)$ are independent, then $X + Y \sim \mathcal{N}(\mu_1 + \mu_2, \Sigma_1 + \Sigma_2)$.

Conditional Distributions

The most important property for Gaussian processes is the form of conditional distributions.

Theorem 7.2 (Gaussian Conditionals)**.** *Let $(X_1, X_2) \sim \mathcal{N}(\mu, \Sigma)$ with partition as above. The conditional distribution of X_1 given $X_2 = x_2$ is:*

$$X_1 \mid X_2 = x_2 \;\sim\; \mathcal{N}(\mu_{1|2},\ \Sigma_{1|2})$$

where:

$$\begin{aligned} \mu_{1|2} &= \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2) \\ \Sigma_{1|2} &= \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \end{aligned}$$

Proof. We derive this by completing the square in the joint log density. The joint log density, up to constants, is:

$$-\frac{1}{2}\begin{pmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{pmatrix}^\top \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}^{-1} \begin{pmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{pmatrix}$$

Using the block matrix inverse formula, the inverse of Σ has the $(1,1)$ block equal to $\Sigma_{1|2}^{-1}$ and the $(1,2)$ block equal to $-\Sigma_{1|2}^{-1}\Sigma_{12}\Sigma_{22}^{-1}$. After substituting and collecting terms in x_1, the quadratic in x_1 is:

$$-\frac{1}{2}(x_1 - \mu_{1|2})^\top \Sigma_{1|2}^{-1}(x_1 - \mu_{1|2}) + \text{terms not involving } x_1$$

The conditional density $p(x_1 \mid x_2) \propto p(x_1, x_2)$ therefore has a quadratic log in x_1, confirming it is Gaussian with the stated mean and covariance. □

The conditional mean formula $\mu_{1|2} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2)$ deserves careful interpretation. The term $\Sigma_{12}\Sigma_{22}^{-1}$ is the **regression coefficient matrix** of X_1 on X_2: it maps deviations of X_2 from its mean to adjustments in the predicted mean of X_1. If X_1 and X_2 are uncorrelated ($\Sigma_{12} = 0$), observing X_2 provides no information about X_1 and the conditional mean equals the prior mean. If Σ_{12} is large, the adjustment can be substantial.

The conditional variance $\Sigma_{1|2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$ is always smaller than (or equal to) the prior variance Σ_{11}: observing X_2 reduces uncertainty about X_1. The term subtracted, $\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$, is the **explained variance**: the reduction in uncertainty achieved by observing X_2.

Example 7.3 (Bivariate Gaussian conditional)**.** Let $(X, Y)^\top \sim \mathcal{N}(0, \Sigma)$ with:

$$\Sigma = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$$

for correlation $\rho \in (-1, 1)$. Then:

$$\mu_{X|Y=y} = 0 + \rho \cdot 1^{-1} \cdot (y - 0) = \rho y$$
$$\sigma^2_{X|Y} = 1 - \rho \cdot 1^{-1} \cdot \rho = 1 - \rho^2$$

So $X \mid Y = y \sim \mathcal{N}(\rho y, 1 - \rho^2)$. When $\rho = 0$, knowing Y tells us nothing about X: the conditional equals the marginal. When $|\rho| \to 1$, the conditional mean is $\pm y$ and the conditional variance vanishes: X is determined by Y.

Example 7.4 (Three variables: indirect information)**.** Let $(X_1, X_2, X_3)^\top \sim \mathcal{N}(0, \Sigma)$ with:

$$\Sigma = \begin{pmatrix} 1 & 0.8 & 0.6 \\ 0.8 & 1 & 0.5 \\ 0.6 & 0.5 & 1 \end{pmatrix}$$

Suppose we observe $X_2 = 1.5$ and $X_3 = -0.5$. The conditional mean of X_1 is:

$$\mu_{1|23} = \begin{pmatrix} 0.8 & 0.6 \end{pmatrix} \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1.5 \\ -0.5 \end{pmatrix}$$

Computing the 2×2 inverse: $\begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}^{-1} = \frac{1}{0.75} \begin{pmatrix} 1 & -0.5 \\ -0.5 & 1 \end{pmatrix}$

Therefore:

$$\mu_{1|23} = \frac{1}{0.75} \begin{pmatrix} 0.8 & 0.6 \end{pmatrix} \begin{pmatrix} 1 & -0.5 \\ -0.5 & 1 \end{pmatrix} \begin{pmatrix} 1.5 \\ -0.5 \end{pmatrix}$$
$$= \frac{1}{0.75}(0.8 \times 1.75 + 0.6 \times (-1.25)) = \frac{1.4 - 0.75}{0.75} \approx 0.867$$

The observation of both X_2 and X_3 updates our belief about X_1, even though X_3 is not directly as correlated with X_1 as X_2: all available information is combined via the conditional formula.

7.3 Gaussian Processes

A **Gaussian process** is the infinite-dimensional generalization of the multivariate Gaussian. Instead of placing a Gaussian distribution over a finite vector, we place one over a function.

Definition 7.5 (Gaussian Process)**.** A **Gaussian process** is a collection of random variables $\{f(x) : x \in \mathcal{X}\}$, one for each point in the input space $\mathcal{X}$, such that for any finite collection of inputs $x_1, \ldots, x_n \in \mathcal{X}$:

$$(f(x_1), \ldots, f(x_n))^\top \sim \mathcal{N}(\boldsymbol{\mu}, K)$$

where $\boldsymbol{\mu}_i = \mu(x_i)$ and $K_{ij} = k(x_i, x_j)$ for a **mean function** $\mu : \mathcal{X} \to \mathbb{R}$ and a **kernel function** (covariance function) $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. We write $f \sim \mathcal{GP}(\mu, k)$.

The kernel $k(x, x')$ encodes our beliefs about the function's smoothness and structure. The key property is: $k(x, x') = \mathrm{Cov}(f(x), f(x'))$ measures how correlated the function values at x and x' are expected to be. If x and x' are nearby, $f(x)$ and $f(x')$ should be similar, so $k(x, x')$ should be large. If x and x' are far apart, $k(x, x')$ should be small.

The GP prior is not a prior over a finite set of parameters but over the entire space of functions. Yet by the definition, working with any finite collection of function values requires only the multivariate Gaussian machinery we just developed.

Common Kernels

Squared exponential (SE) kernel:

$$k_{\mathrm{SE}}(x, x') = \sigma_f^2 \exp\left(-\frac{(x - x')^2}{2\ell^2}\right)$$

The parameters are the **output scale** σ_f^2 (overall variance of function values) and the **lengthscale** ℓ (how quickly the function varies). Functions sampled from a GP with this kernel are infinitely differentiable. For $x = x'$: $k(x, x) = \sigma_f^2$, the prior variance at any single point.

Matérn kernel: A family of kernels parameterized by a smoothness parameter ν. The Matérn-5/2 kernel:

$$k_{\mathrm{M52}}(x, x') = \sigma_f^2 \left(1 + \frac{\sqrt{5}|x - x'|}{\ell} + \frac{5(x - x')^2}{3\ell^2}\right) \exp\left(-\frac{\sqrt{5}|x - x'|}{\ell}\right)$$

produces functions that are twice differentiable — a common choice for physical systems that are smooth but not infinitely so.

Periodic kernel:

$$k_{\mathrm{per}}(x, x') = \sigma_f^2 \exp\left(-\frac{2\sin^2(\pi(x - x')/p)}{\ell^2}\right)$$

produces periodic functions with period p.

For the rest of this chapter we use the squared exponential kernel with $\sigma_f^2 = 1$ and ℓ to be specified.

GP Regression: The Posterior

We observe noisy function values at n training points $X = (x_1, \dots, x_n)^\top$:

$$y_i = f(x_i) + \varepsilon_i, \quad \varepsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$$

We want the posterior distribution of f at a new test point x_*. Using the GP prior $f \sim \mathcal{GP}(0, k)$ (zero mean for simplicity), the joint distribution of the training values and the test value is Gaussian:

$$\begin{pmatrix} y \\ f(x_*) \end{pmatrix} \sim \mathcal{N}\left(0,\ \begin{pmatrix} K(X,X) + \sigma^2 I & k(X, x_*) \\ k(x_*, X) & k(x_*, x_*) \end{pmatrix}\right)$$

where:

- $K(X, X) \in \mathbb{R}^{n \times n}$ is the **kernel matrix** with entries $[K(X,X)]_{ij} = k(x_i, x_j)$
- $k(X, x_*) \in \mathbb{R}^n$ is the vector of kernel evaluations between training points and test point: $[k(X, x_*)]_i = k(x_i, x_*)$
- $k(x_*, X) = k(X, x_*)^\top$

Applying Theorem 7.2 with $X_1 = f(x_*)$ and $X_2 = y$:

GP Posterior at a single test point x_*:

$$f(x_*) \mid y \ \sim \ \mathcal{N}(\mu_*, \sigma_*^2)$$

where:

$$\mu_* = k(x_*, X)\left(K(X,X) + \sigma^2 I\right)^{-1} y$$
$$\sigma_*^2 = k(x_*, x_*) - k(x_*, X)\left(K(X,X) + \sigma^2 I\right)^{-1} k(X, x_*)$$

The posterior mean μ_* is a weighted combination of the training observations y, where the weights $k(x_*, X)(K(X,X) + \sigma^2 I)^{-1}$ depend on how similar x_* is to each training point. Points similar to x_* (large $k(x_*, x_i)$) receive more weight; dissimilar points receive less.

The posterior variance σ_*^2 is the prior variance $k(x_*, x_*)$ minus the **explained variance** $k(x_*, X)(K(X,X)+\sigma^2 I)^{-1}k(X, x_*)$. The posterior variance is small near training points and large in unexplored regions — exactly as we want for a model that is confident where it has data and uncertain where it does not.

For a set of test points $X_* = (x_*^{(1)}, \ldots, x_*^{(m)})^\top$, the posterior is jointly Gaussian:

$$f(X_*) \mid y \;\sim\; \mathcal{N}(\mu_{X_*}, \Sigma_{X_*})$$

where:

$$\mu_{X_*} = K(X_*, X)(K(X, X) + \sigma^2 I)^{-1} y$$
$$\Sigma_{X_*} = K(X_*, X_*) - K(X_*, X)(K(X, X) + \sigma^2 I)^{-1} K(X, X_*)$$

Example 7.6 (GP posterior with three observations)**.** Let $\mathcal{X} = [0, 5]$, kernel $k(x, x') = \exp(-(x - x')^2/2)$ (SE with $\ell = 1$, $\sigma_f^2 = 1$), and noise $\sigma^2 = 0.1$. Three observations:

$$x_1 = 1,\ y_1 = 0.5; \quad x_2 = 2.5,\ y_2 = 2.0; \quad x_3 = 4,\ y_3 = 1.2$$

The kernel matrix:

$$K(X, X) = \begin{pmatrix} k(1,1) & k(1,2.5) & k(1,4) \\ k(2.5,1) & k(2.5,2.5) & k(2.5,4) \\ k(4,1) & k(4,2.5) & k(4,4) \end{pmatrix} = \begin{pmatrix} 1.000 & 0.325 & 0.011 \\ 0.325 & 1.000 & 0.325 \\ 0.011 & 0.325 & 1.000 \end{pmatrix}$$

The noisy kernel matrix $(K + 0.1I)$:

$$K + \sigma^2 I = \begin{pmatrix} 1.100 & 0.325 & 0.011 \\ 0.325 & 1.100 & 0.325 \\ 0.011 & 0.325 & 1.100 \end{pmatrix}$$

At a test point $x_* = 3$, the kernel vector is:

$$k(X, x_*) = \begin{pmatrix} k(1,3) \\ k(2.5,3) \\ k(4,3) \end{pmatrix} = \begin{pmatrix} e^{-2} \\ e^{-0.125} \\ e^{-0.5} \end{pmatrix} \approx \begin{pmatrix} 0.135 \\ 0.882 \\ 0.607 \end{pmatrix}$$

The posterior mean and variance at $x_* = 3$ are computed by the formulas above. The test point $x_* = 3$ is closest to $x_2 = 2.5$ and $x_3 = 4$, so the posterior mean is a weighted average of $y_2 = 2.0$ and $y_3 = 1.2$, pulled slightly toward 1.2 by the observation at x_3. The posterior variance at $x_* = 3$ is moderate — this point is between two training observations and so is reasonably well constrained, but not as tightly as a point directly at a training location would be.

The posterior variance at $x_* = 2.5$ (a training point) is approximately $\sigma^2 = 0.1$: equal to the noise level, since the observation there is noisy and the function value is not exactly pinned. At a point far from all training data, say $x_* = 5$, the posterior variance approaches the prior variance $k(5, 5) = 1$: no data, maximum uncertainty.

7.4 Bayesian Optimization

We now turn to the optimization problem. We have an unknown black-box function $f : \mathcal{X} \to \mathbb{R}$ and want to find:

$$x^* = \arg\max_{x \in \mathcal{X}} f(x)$$

We can evaluate f at chosen points, but each evaluation is expensive. We want to find x^* (or at least a good approximation) using as few evaluations as possible.

The challenge is the **exploration-exploitation dilemma**:

- **Exploitation:** evaluate where the posterior mean is highest — where we currently expect f to be largest. Risk: the true maximum may be elsewhere, in a region we have not explored.
- **Exploration:** evaluate where posterior uncertainty is highest — regions we know little about. Risk: these regions may have low f values and contribute nothing to finding the maximum.

Without a principled framework, balancing these is a matter of heuristics. The Bayesian approach resolves this by turning the decision into an optimization problem: at each step, choose the next evaluation point by maximizing an **acquisition function** that explicitly trades off mean and uncertainty.

The Bayesian Optimization Algorithm

Bayesian Optimization Algorithm:

1. **Initialize.** Evaluate f at a small number of initial points (typically 3–10), obtaining observations $\mathcal{D}_0 = \{(x_i, y_i)\}_{i=1}^{n_0}$.
2. **Fit GP.** Compute the GP posterior given current observations $\mathcal{D}_t$: the posterior mean $\mu_t(x)$ and variance $\sigma_t^2(x)$ at every point x.
3. **Maximize acquisition.** Choose the next evaluation point:

   $$x_{t+1} = \arg\max_{x \in \mathcal{X}} \alpha(x; \mathcal{D}_t)$$

   where α is the acquisition function.
4. **Evaluate.** Observe $y_{t+1} = f(x_{t+1}) + \varepsilon_{t+1}$.
5. **Update.** Add (x_{t+1}, y_{t+1}) to $\mathcal{D}_t$ to form $\mathcal{D}_{t+1}$. Return to step 2.

Repeat until the evaluation budget is exhausted. Return $x^* = \arg\max_{x \in \{x_1, \ldots, x_T\}} y_i$ as the estimated optimum.

Acquisition Functions

Expected Improvement (EI). Let $f^+ = \max_{i \le t} y_i$ be the best function value observed so far. The **improvement** at x is:

$$I(x) = \max(f(x) - f^+, 0)$$

This is zero if $f(x) \le f^+$ (no improvement) and positive otherwise. The expected improvement is:

$$\mathrm{EI}(x) = \mathbb{E}[I(x) \mid \mathcal{D}_t] = \mathbb{E}[\max(f(x) - f^+, 0) \mid \mathcal{D}_t]$$

Since $f(x) \mid \mathcal{D}_t \sim \mathcal{N}(\mu_t(x), \sigma_t^2(x))$, let $Z = (f(x) - f^+)/\sigma_t(x) \sim \mathcal{N}(0, 1)$. Then:

$$\begin{aligned}\mathrm{EI}(x) &= \mathbb{E}\big[\max\big(\mu_t(x) + \sigma_t(x)Z - f^+,\, 0\big)\big] \\ &= (\mu_t(x) - f^+)\,\Phi\left(\frac{\mu_t(x) - f^+}{\sigma_t(x)}\right) + \sigma_t(x)\,\phi\left(\frac{\mu_t(x) - f^+}{\sigma_t(x)}\right)\end{aligned}$$

where Φ is the standard normal CDF and ϕ is the standard normal PDF.

Define $\gamma(x) = (\mu_t(x) - f^+)/\sigma_t(x)$ (the "standardized improvement"). Then:

$$\mathrm{EI}(x) = (\mu_t(x) - f^+)\,\Phi(\gamma(x)) + \sigma_t(x)\,\phi(\gamma(x))$$

The two terms have clean interpretations:

- $(\mu_t(x) - f^+)\Phi(\gamma(x))$: the expected amount by which $f(x)$ exceeds f^+, weighted by the probability of improvement. This is the **exploitation** term: large when $\mu_t(x)$ is large.
- $\sigma_t(x)\phi(\gamma(x))$: a positive correction for uncertainty. Even when $\mu_t(x) < f^+$, if $\sigma_t(x)$ is large, there is a reasonable chance $f(x) > f^+$. This is the **exploration** term: large when $\sigma_t(x)$ is large.

Upper Confidence Bound (UCB). An alternative acquisition function:

$$\mathrm{UCB}(x) = \mu_t(x) + \kappa\,\sigma_t(x)$$

where $\kappa > 0$ controls the exploration-exploitation tradeoff. Maximizing UCB selects the point with the highest optimistic estimate of $f(x)$: the posterior mean plus κ standard deviations. The parameter κ is a hyperparameter; larger κ means more exploration. Theoretical results by Srinivas et al. (2010) provide regret bounds for UCB with an appropriately chosen κ.

Probability of Improvement (PI).

$$\mathrm{PI}(x) = P(f(x) > f^+ \mid \mathcal{D}_t) = \Phi(\gamma(x))$$

This is the probability that the new point improves on the current best. PI is simple but tends to be more greedy than EI: it maximizes the chance of any improvement rather than the expected magnitude, so it can get stuck near the current best.

In practice, EI is the most widely used acquisition function because it balances exploration and exploitation without requiring manual tuning of κ.

7.5 A Concrete Example: Optimizing a Black-Box Function

We work through a complete BO run in detail. The black-box function is:

$$f(x) = \sin(3x) + x\cos(x) - \frac{x^2}{8}, \qquad x \in [0, 6]$$

This function is multimodal (several local maxima) and has a global maximum near $x \approx 2.1$ with $f(2.1) \approx 2.08$. We pretend we do not know the formula and observe only noisy evaluations $y = f(x) + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, 0.1^2)$.

GP setup. We use:

- Kernel: $k(x, x') = \exp(-(x - x')^2/(2 \times 1.0^2))$ (SE kernel with $\ell = 1.0$, $\sigma_f^2 = 1$)
- Noise: $\sigma^2 = 0.1^2 = 0.01$

We walk through the first five iterations in detail.

Iteration 0: Initial Observations

We start with two observations at the endpoints:

$$x_1 = 0.5,\ y_1 = f(0.5) + \varepsilon_1 \approx 0.44$$

$$x_2 = 5.0,\ y_2 = f(5.0) + \varepsilon_2 \approx -0.88$$

These two points give the GP a rough sense of the function's scale. The posterior mean passes through both observations, and the posterior variance is high everywhere else.

The current best: $f^+ = 0.44$ at $x = 0.5$.

Iteration 1: Compute Posterior and EI

With two observations, we compute the 2×2 kernel matrix:

$$K(X, X) = \begin{pmatrix} k(0.5, 0.5) & k(0.5, 5.0) \\ k(5.0, 0.5) & k(5.0, 5.0) \end{pmatrix}$$

$$= \begin{pmatrix} 1.000 & e^{-(4.5)^2/2} \\ e^{-(4.5)^2/2} & 1.000 \end{pmatrix} = \begin{pmatrix} 1.000 & 0.000 \\ 0.000 & 1.000 \end{pmatrix}$$

(The two points are so far apart that $k(0.5, 5.0) = e^{-10.125} \approx 0$; they carry no information about each other.) The noisy kernel matrix is $(K + 0.01I)$.

At a candidate point $x = 2.0$:

$$k(X, 2.0) = \begin{pmatrix} k(0.5, 2.0) \\ k(5.0, 2.0) \end{pmatrix} = \begin{pmatrix} e^{-1.125} \\ e^{-4.5} \end{pmatrix} \approx \begin{pmatrix} 0.325 \\ 0.011 \end{pmatrix}$$

The posterior mean at $x = 2.0$:

$$\mu_1(2.0) \approx \begin{pmatrix} 0.325 & 0.011 \end{pmatrix} (K + 0.01I)^{-1} \begin{pmatrix} 0.44 \\ -0.88 \end{pmatrix} \approx 0.141$$

The posterior variance at $x = 2.0$:

$$\sigma_1^2(2.0) = 1 - \begin{pmatrix} 0.325 & 0.011 \end{pmatrix} (K + 0.01I)^{-1} \begin{pmatrix} 0.325 \\ 0.011 \end{pmatrix} \approx 0.895$$

So $\sigma_1(2.0) \approx 0.946$: high uncertainty here, since $x = 2.0$ is far from both training points.

Computing EI at $x = 2.0$ with $f^+ = 0.44$:

$$\gamma = \frac{0.141 - 0.44}{0.946} = \frac{-0.299}{0.946} \approx -0.316$$

$$\text{EI}(2.0) = (0.141 - 0.44)\,\Phi(-0.316) + 0.946\,\phi(-0.316)$$

$$= (-0.299)(0.376) + 0.946 \times 0.378 \approx -0.112 + 0.357 = 0.245$$

Despite the posterior mean being below the current best, the high uncertainty gives EI a substantial positive value. The exploration term dominates.

We compute $\text{EI}(x)$ on a fine grid of x values across $[0, 6]$. The maximum of EI occurs at approximately $x = 1.8$, where the posterior mean is moderate and the uncertainty is high.

Next evaluation: $x_3 = 1.8$, $y_3 = f(1.8) + \varepsilon_3 \approx 1.87$.

Update current best: $f^+ = 1.87$ at $x_3 = 1.8$.

Iteration 2

Now we have three observations: $(0.5, 0.44)$, $(5.0, -0.88)$, $(1.8, 1.87)$. The new observation at $x = 1.8$ is much higher than the previous best. The posterior mean is now pulled upward near $x = 1.8$, and the GP has learned that the function is high in the $[1.5, 2.5]$ region.

Computing the 3×3 kernel matrix and posterior:

Point	$\mu_2(x)$	$\sigma_2(x)$	$\text{EI}(x)$
$x = 1.0$	0.93	0.71	0.282
$x = 2.0$	1.61	0.52	0.314
$x = 2.5$	1.12	0.68	0.161
$x = 3.5$	0.31	0.88	0.193
$x = 4.5$	-0.52	0.63	0.002

The maximum EI is at approximately $x = 2.0$, where the posterior mean is high (exploitation) and uncertainty is still moderate (some exploration). The region near $x = 4.5$ has low EI despite moderate uncertainty because the posterior mean is very low there: the data has already revealed that this region is unpromising.

Next evaluation: $x_4 = 2.0$, $y_4 = f(2.0) + \varepsilon_4 \approx 2.01$.

Update current best: $f^+ = 2.01$ at $x_4 = 2.0$.

Iteration 3

Four observations: $(0.5, 0.44)$, $(1.8, 1.87)$, $(2.0, 2.01)$, $(5.0, -0.88)$. The two nearby high-value observations at $x = 1.8$ and $x = 2.0$ give the GP strong evidence that the maximum is in the $[1.5, 2.5]$ region. The posterior mean peaks around $x \approx 2.0$–2.1 and falls off on either side.

The posterior variance is now quite small near $x \in [1.5, 2.5]$ (two nearby observations constrain the function tightly) but remains large in unexplored regions like $[2.5, 4.5]$.

EI now balances two competing demands:

- **Exploit near the current best:** evaluate near $x = 2.0$–2.1, where the posterior mean is highest. But the variance there is small, so EI's exploitation term is not as dominant.
- **Explore the gap** $[2.5, 4.5]$**:** high uncertainty here. But the posterior mean is moderate or low, making the exploration term contribute only if there is real probability of exceeding $f^+ = 2.01$.

The maximum EI falls at approximately $x = 2.1$, slightly to the right of the current best, refining the estimate of the maximum's location.

Next evaluation: $x_5 = 2.1$, $y_5 = f(2.1) + \varepsilon_5 \approx 2.09$.

Update current best: $f^+ = 2.09$ at $x_5 = 2.1$.

Iteration 4

Five observations. The region near $x = 2.1$ is now tightly constrained. The posterior mean peaks very close to the true maximum at $x \approx 2.1$. The posterior variance there is small.

Iter	New point	y	f^+	Location of f^+
0	$x_1 = 0.5$	0.44	0.44	$x = 0.5$
0	$x_2 = 5.0$	-0.88	0.44	$x = 0.5$
1	$x_3 = 1.8$	1.87	1.87	$x = 1.8$
2	$x_4 = 2.0$	2.01	2.01	$x = 2.0$
3	$x_5 = 2.1$	2.09	2.09	$x = 2.1$
4	$x_6 = 2.1$	2.07	2.09	$x = 2.1$

After 6 evaluations, the algorithm has found the global maximum to within 0.01 of its true location. A grid search over $[0, 6]$ with the same 6 evaluations would place points at $x = 0, 1.2, 2.4, 3.6, 4.8, 6.0$, missing the true maximum at $x = 2.1$ entirely. The BO algorithm directed all its evaluations toward the promising region through principled reasoning under uncertainty.

Why EI Works: The Detailed Mechanism

Looking back at the five iterations, the mechanism of EI is clear at each step:

Early iterations (high uncertainty everywhere): the exploration term $\sigma_t(x)\phi(\gamma(x))$ dominates. EI directs evaluations toward unexplored regions regardless of the posterior mean, because uncertainty is the primary driver of potential improvement.

Middle iterations (uncertainty concentrated in gaps): both terms matter. EI evaluates near high-mean, high-variance regions: places the model thinks are probably good but is not yet certain about.

Late iterations (local region well-explored): the exploitation term $(\mu_t(x) - f^+)\Phi(\gamma(x))$ dominates near the current best, and EI refines the estimate of the maximum location. In unpromising regions (low mean), EI is small even if variance is high, because the chance of improving on f^+ is small.

This automatic transition from exploration to exploitation is not programmed explicitly — it falls out of the EI formula and the GP posterior together. The GP uncertainty quantification is doing the essential work.

Why Bayesian Optimization is not just gradient-free optimization. Many gradient-free optimizers (Nelder-Mead, CMA-ES, random search) do not model uncertainty. They make no explicit attempt to reason about where the function might be high versus where it is known to be low. BO's advantage is that the GP posterior tracks both what is known and what is unknown, and the acquisition function makes decisions based on both. This is what Bayesian reasoning enables that classical optimization does not: principled sequential decision making under uncertainty, with the uncertainty quantification doing essential computational work at every step.

7.6 Practical Considerations

Choosing the Kernel

The kernel determines the function space the GP explores. A misspecified kernel can lead to poor optimization:

- Too small a lengthscale ℓ: the GP assumes very rough functions. Each observation has little influence on nearby points, so the GP never generalizes and EI directs many evaluations to the same local region.

- Too large a lengthscale ℓ: the GP assumes very smooth functions. The posterior mean is oversmoothed and may miss the true maximum.

In practice, the kernel hyperparameters $(\ell, \sigma_f^2, \sigma^2)$ are estimated by maximizing the **marginal likelihood**:

$$\log p(y \mid X, \ell, \sigma_f^2, \sigma^2) = -\frac{1}{2} y^\top (K + \sigma^2 I)^{-1} y - \frac{1}{2} \log |K + \sigma^2 I| - \frac{n}{2} \log(2\pi)$$

This is the Bayesian model evidence: the probability of the observed data under the GP prior. Maximizing it over hyperparameters selects the kernel that best explains the data. The gradient is available analytically, so this optimization is done by gradient ascent.

High-Dimensional Inputs

The SE kernel in d dimensions is:

$$k(x, x') = \sigma_f^2 \exp\left(-\frac{1}{2} \sum_{j=1}^{d} \frac{(x_j - x'_j)^2}{\ell_j^2} \right)$$

with a separate lengthscale ℓ_j per dimension (**automatic relevance determination**, ARD). Dimensions with large ℓ_j are irrelevant: the function does not vary much along them. ARD automatically performs variable selection: the BO iterates reveal which input dimensions matter.

Computational Cost

Each GP posterior update requires inverting the $(n \times n)$ matrix $(K + \sigma^2 I)$, which costs $O(n^3)$ by Cholesky decomposition. This becomes expensive for $n \gtrsim 1000$. Approximate GP methods (sparse GPs, random features, inducing point methods) reduce this to $O(nm^2)$ or $O(n \log n)$ for $m \ll n$ inducing points, enabling BO at larger scales.

Exercise 7.7. Let $f(x) = -x^2 + 4x - 3$ on $[0, 4]$ (a quadratic with maximum at $x = 2$, $f(2) = 1$). Use the SE kernel with $\ell = 1$, $\sigma_f^2 = 1$, $\sigma^2 = 0.01$.

1. Starting with observations $x_1 = 0.5$, $y_1 = -1.5$ and $x_2 = 3.5$, $y_2 = -1.0$, compute the 2×2 kernel matrix $K(X, X) + \sigma^2 I$.

2. Compute the posterior mean $\mu_1(x)$ and variance $\sigma_1^2(x)$ at $x = 1, 1.5, 2, 2.5, 3$.

3. Compute $\mathrm{EI}(x)$ at these five points with $f^+ = \max(-1.5, -1.0) = -1.0$. Which point has the highest EI?

4. Evaluate f at the point with highest EI. Update the posterior and compute EI again. Does BO move toward the true maximum at $x = 2$?

Exercise 7.8. Derive the EI formula $\mathrm{EI}(x) = (\mu_t(x) - f^+)\Phi(\gamma(x)) + \sigma_t(x)\phi(\gamma(x))$ by computing $\mathbb{E}[\max(Z - \gamma, 0)]$ for $Z \sim \mathcal{N}(0,1)$, where $\gamma = (f^+ - \mu_t(x))/\sigma_t(x)$. (Hint: split the integral at $Z = \gamma$ and use the identity $\int_\gamma^\infty z\,\phi(z)\,dz = \phi(\gamma)$.)

Exercise 7.9. Show that for a two-point GP with observations (x_1, y_1) and (x_2, y_2), the posterior mean at a test point x_* is:

$$\mu_*(x_*) = \alpha_1 k(x_*, x_1) + \alpha_2 k(x_*, x_2)$$

for some weights α_1, α_2 that depend on the observations and kernel matrix. Compute α_1 and α_2 explicitly using the 2×2 matrix inverse formula, and verify that $\mu_*(x_1) \approx y_1$ and $\mu_*(x_2) \approx y_2$ (approximately, due to noise). Show that the posterior mean is a linear combination of kernel functions centered at the training points — a fact that holds in general and is called the **representer theorem**.

7.7 Marginal Likelihood and Hyperparameter Tuning

Every model has hyperparameters: quantities that are not estimated from the likelihood directly but that control the structure of the model. In Gaussian process regression, the hyperparameters are the kernel parameters: the lengthscale ℓ, the output scale σ_f^2, and the noise variance σ^2. These are not the function values $f(x_*)$ that the GP predicts; they are the parameters of the prior over functions. Choosing them well is essential. A lengthscale that is too small makes the GP assume a very rough function; one that is too large makes it assume an unrealistically smooth one. Neither gives good predictions.

The Standard Answer: Cross-Validation

The standard frequentist approach to hyperparameter selection is **cross-validation**. Split the n training observations into K folds. For each candidate hyperparameter setting $\psi = (\ell, \sigma_f^2, \sigma^2)$:

1. Hold out fold k as a validation set.
2. Fit the GP on the remaining $K - 1$ folds.
3. Evaluate prediction error on the held-out fold.
4. Average the error over all K folds.

Select the ψ that minimizes the average cross-validation error. The most common choice is $K = 5$ or $K = 10$; leave-one-out (LOO) cross-validation uses $K = n$.

Cross-validation has genuine virtues. It is model-agnostic: it works for any prediction method, not just GPs. It estimates the actual prediction error on held-out

data, which is the quantity practitioners care about. And it makes no distributional assumptions beyond those of the model itself.

But it has real costs:

- **Computational cost.** Each fold requires fitting a separate model. For a GP with n observations, each fit costs $O(n^3)$ for the matrix inversion. With $K = 10$ folds and a grid of M hyperparameter settings, the total cost is $O(10Mn^3)$. For a GP with $n = 500$ and a $10 \times 10 \times 10$ grid of hyperparameters, this is 10,000 separate matrix inversions.

- **Data waste.** Each fold withholds n/K observations from training. In small datasets, this can substantially degrade the quality of the fit used for evaluation.

- **Variance.** The cross-validation estimate of prediction error is itself random, with variance that depends on the correlation between folds. The selected hyperparameters may not generalize well to truly new data.

- **No uncertainty.** Cross-validation selects a single hyperparameter setting. It does not quantify uncertainty about whether this setting is correct.

The Bayesian approach provides an alternative that avoids all four problems simultaneously.

The Marginal Likelihood

The **marginal likelihood** (also called the **evidence**) is the probability of the observed data under the model, with the function values f integrated out:

$$p(y \mid X, \psi) = \int p(y \mid f, X)\, p(f \mid X, \psi)\, df$$

For a GP with kernel k_ψ, this integral is analytically tractable because both the prior $p(f \mid X, \psi)$ and the likelihood $p(y \mid f, X) = \mathcal{N}(y; f, \sigma^2 I)$ are Gaussian. The result is:

$$p(y \mid X, \psi) = \mathcal{N}(y;\ 0,\ K_\psi + \sigma^2 I)$$

where K_ψ is the $n \times n$ kernel matrix with entries $[K_\psi]_{ij} = k_\psi(x_i, x_j)$. The log marginal likelihood is:

$$\begin{aligned} \log p(y \mid X, \psi) = &-\frac{1}{2} y^\top (K_\psi + \sigma^2 I)^{-1} y \\ &- \frac{1}{2} \log |K_\psi + \sigma^2 I| \\ &- \frac{n}{2} \log(2\pi) \end{aligned}$$

This is computable in $O(n^3)$ — the cost of a single Cholesky decomposition of the $n \times n$ matrix $(K_\psi + \sigma^2 I)$. Its gradient with respect to ψ is also analytically available:

$$\frac{\partial}{\partial \psi_j} \log p(y \mid X, \psi) = \frac{1}{2} \mathrm{tr}\left((\alpha\alpha^\top - C^{-1}) \frac{\partial K_\psi}{\partial \psi_j} \right)$$

where $C = K_\psi + \sigma^2 I$ and $\alpha = C^{-1} y$. This gradient enables optimization of the marginal likelihood by gradient ascent, using the *same* Cholesky decomposition that computes the marginal likelihood itself. The total cost of one gradient step is $O(n^3)$ — a single matrix inversion, not K of them.

What the Three Terms Mean

The log marginal likelihood decomposes into three terms with clean interpretations:

$$\log p(y \mid X, \psi) = \underbrace{-\frac{1}{2} y^\top C^{-1} y}_{\text{data fit}} \quad \underbrace{-\frac{1}{2} \log |C|}_{\text{complexity penalty}} \quad \underbrace{-\frac{n}{2} \log(2\pi)}_{\text{constant}}$$

Data fit term. The quadratic form $y^\top C^{-1} y$ measures how well the GP prior, with kernel k_ψ, explains the observed data. A model that fits the data well has small $y^\top C^{-1} y$.

Complexity penalty term. The log determinant $\log |C|$ penalizes model complexity. A more flexible model (larger ℓ^{-1} or σ_f^2) has a larger C, a larger determinant, and hence a larger penalty. This is the Bayesian Occam's razor: more complex models are penalized for spreading their prior probability over a larger space of functions.

Automatic tradeoff. These two terms act in opposition. Increasing σ_f^2 (a more flexible prior) improves the data fit but increases the complexity penalty. The maximum of the marginal likelihood is the point where the marginal gain in fit from more flexibility is exactly offset by the marginal increase in complexity penalty. This balance happens automatically, without any held-out data.

A Concrete Example

Consider $n = 20$ observations from the function $f(x) = \sin(2\pi x)$ with noise $\sigma^2 = 0.1$, on the interval $[0, 1]$. We use a squared exponential kernel $k(x, x') = \sigma_f^2 \exp(-(x - x')^2/(2\ell^2))$ and optimize $\psi = (\ell, \sigma_f^2, \sigma^2)$ by maximizing the log marginal likelihood.

ℓ	σ_f^2	σ^2	$\log p(y \mid X, \psi)$	Behavior
0.05	1.0	0.1	-34.2	Too rough: overfits noise
0.20	1.0	0.1	-18.6	About right
0.18	0.92	0.09	-17.1	Optimum (gradient ascent)
2.00	1.0	0.1	-29.8	Too smooth: underfits

The optimized $\hat{\ell} = 0.18$ is close to the true lengthscale of $\sin(2\pi x)$, which varies on a scale of $\sim 1/(2\pi) \approx 0.16$. The optimization took one gradient ascent run on 20 data points: a single 20×20 matrix inversion per gradient step, compared to the $10 \times K \times$ matrix inversions that cross-validation would require.

At the extremes:

- Small $\ell = 0.05$: the kernel assumes a very rough function. The GP can fit any dataset, including pure noise. The data fit term is excellent, but the complexity penalty is severe because the prior puts mass on an enormous space of rough functions.

- Large $\ell = 2.00$: the kernel assumes a nearly constant function. The GP cannot reproduce the oscillations of $\sin(2\pi x)$. The data fit term is poor, and even though the complexity penalty is low, the total marginal likelihood is low.

Marginal Likelihood vs. Cross-Validation: A Direct Comparison

Property	**Cross-validation**	**Marginal likelihood**
Computational cost	$O(KMn^3)$: K fits per candidate	$O(n^3)$: one fit, gradient ascent
Uses all data for fitting	No: withholds n/K per fold	Yes: all n points used
Provides uncertainty	No: selects single ψ	Can integrate over ψ
Automatic complexity penalty	No: must specify K	Yes: built into log determinant
Requires held-out data	Yes	No
Model-agnostic	Yes	No: requires tractable integral
LOO equivalent	Leave-one-out CV	Approximately equivalent for GPs

The last row deserves emphasis. For Gaussian process regression specifically, it can be shown that the leave-one-out cross-validation score is closely related to the marginal likelihood: both measure predictive performance on unseen data, and for large n they select similar hyperparameters. The marginal likelihood achieves this at a fraction of the computational cost.

When cross-validation wins. Cross-validation is model-agnostic: it works for random forests, neural networks, and any black-box predictor where the marginal likelihood integral is intractable. For GPs specifically, the marginal likelihood is the superior tool — analytically available, computationally cheaper, and theoretically better motivated.

Full Bayes: Integrating Over Hyperparameters

Maximizing the marginal likelihood selects a single hyperparameter setting $\hat{\psi}$. This is still a point estimate — an empirical Bayes approach (like the James-Stein estimator of Chapter 8). The fully Bayesian treatment places a hyperprior $p(\psi)$ on the hyperparameters and integrates:

$$p(y_* \mid x_*, X, y) = \int p(y_* \mid x_*, X, y, \psi)\, p(\psi \mid X, y)\, d\psi$$

This integral is rarely available in closed form and requires MCMC or further approximation. In practice, maximizing the marginal likelihood is the standard approach for GP hyperparameter tuning, and it works well: the marginal likelihood is typically well-concentrated around $\hat{\psi}$ for moderate n, so integrating over ψ adds little.

7.8 Bayes Factors for Model Selection

The marginal likelihood selects among continuous hyperparameter settings by optimization. The same object can select among discrete model choices by comparison. This is the **Bayes factor**.

The Setup

Suppose we have two competing models $\mathcal{M}_1$ and $\mathcal{M}_2$ for the same data y. In the GP context, $\mathcal{M}_1$ might use a squared exponential kernel and $\mathcal{M}_2$ a Matérn-5/2 kernel. Or $\mathcal{M}_1$ might be a GP with a single lengthscale and $\mathcal{M}_2$ might use a separate lengthscale per input dimension (ARD). Or one model might include a linear trend and the other might not.

Each model has its own marginal likelihood:

$$p(y \mid \mathcal{M}_k) = \int p(y \mid \theta, \mathcal{M}_k)\, p(\theta \mid \mathcal{M}_k)\, d\theta$$

This is the probability of the observed data under model $\mathcal{M}_k$, averaged over all parameter settings weighted by the prior. It is the same marginal likelihood we just computed for the GP, now made explicit about which model it belongs to.

The **Bayes factor** in favor of $\mathcal{M}_1$ over $\mathcal{M}_2$ is:

$$\mathrm{BF}_{12} = \frac{p(y \mid \mathcal{M}_1)}{p(y \mid \mathcal{M}_2)}$$

The Bayes factor is a likelihood ratio for models. Just as the likelihood ratio $p(y \mid \theta_1)/p(y \mid \theta_2)$ compares two parameter values, the Bayes factor compares two models by the probability each assigns to the observed data, with parameters integrated out. A Bayes factor $\mathrm{BF}_{12} = 10$ means the data is ten times more probable under $\mathcal{M}_1$ than under $\mathcal{M}_2$.

From Bayes Factor to Posterior Model Probability

If we place prior probabilities $P(\mathcal{M}_1)$ and $P(\mathcal{M}_2) = 1 - P(\mathcal{M}_1)$ on the two models, Bayes rule gives the posterior model probabilities:

$$P(\mathcal{M}_1 \mid y) = \frac{p(y \mid \mathcal{M}_1)\, P(\mathcal{M}_1)}{p(y \mid \mathcal{M}_1)\, P(\mathcal{M}_1) + p(y \mid \mathcal{M}_2)\, P(\mathcal{M}_2)}$$

$$= \frac{\mathrm{BF}_{12}\, P(\mathcal{M}_1)}{\mathrm{BF}_{12}\, P(\mathcal{M}_1) + P(\mathcal{M}_2)}$$

With equal prior model probabilities $P(\mathcal{M}_1) = P(\mathcal{M}_2) = 1/2$:

$$P(\mathcal{M}_1 \mid y) = \frac{\mathrm{BF}_{12}}{1 + \mathrm{BF}_{12}}$$

A Bayes factor of 10 gives $P(\mathcal{M}_1 \mid y) = 10/11 \approx 91\%$. A Bayes factor of 100 gives 99%. The scale of evidence is:

BF_{12}	$\log_{10} \mathrm{BF}_{12}$	Interpretation
1–3	0–0.5	Barely worth mentioning
3–10	0.5–1	Substantial evidence for $\mathcal{M}_1$
10–100	1–2	Strong evidence
>100	>2	Decisive evidence

This scale (due to Jeffreys, 1961) provides a calibrated vocabulary for describing the strength of evidence. Unlike a p-value, the Bayes factor is symmetric: $\mathrm{BF}_{21} = 1/\mathrm{BF}_{12}$, and the scale of evidence against $\mathcal{M}_1$ mirrors the scale in favor of it.

A Concrete Example: Kernel Selection for a GP

We observe $n = 30$ data points from an unknown function and must choose between:

- $\mathcal{M}_1$: GP with squared exponential kernel $k_1(x, x') = \sigma_f^2 \exp(-(x - x')^2/(2\ell^2))$ — assumes infinitely differentiable functions.
- $\mathcal{M}_2$: GP with Matérn-5/2 kernel $k_2(x, x') = \sigma_f^2(1 + \sqrt{5}r/\ell + 5r^2/(3\ell^2))e^{-\sqrt{5}r/\ell}$ where $r = |x - x'|$ — assumes twice differentiable functions.

- $\mathcal{M}_3$: GP with Matérn-3/2 kernel — assumes once differentiable functions.

For each model, we optimize the hyperparameters $\psi = (\ell, \sigma_f^2, \sigma^2)$ by maximizing the marginal likelihood within that model. The resulting optimized log marginal likelihoods are:

Model	Kernel	$\log p(y \mid \mathcal{M}_k)$	$P(\mathcal{M}_k \mid y)$
$\mathcal{M}_1$	Squared exponential	-24.3	5%
$\mathcal{M}_2$	Matérn-5/2	-21.1	95%
$\mathcal{M}_3$	Matérn-3/2	-28.7	$< 1\%$

The Bayes factor $\text{BF}_{21} = \exp(-21.1 - (-24.3)) = e^{3.2} \approx 24.5$: strong evidence for the Matérn-5/2 kernel. The data is 24 times more probable under $\mathcal{M}_2$ than under $\mathcal{M}_1$. The model comparison has automatically detected that the true function is twice but not infinitely differentiable — a genuine inference about the smoothness of the unknown function.

Notice what happened here. The squared exponential kernel ($\mathcal{M}_1$) can fit smooth functions excellently. But it is *too* flexible: it assigns prior probability to functions far smoother than the data supports. The marginal likelihood penalizes this through the complexity penalty term. The Matérn-5/2 ($\mathcal{M}_2$) matches the actual smoothness of the data and is rewarded.

Variable Selection: Bayes Factors for Regression

Another vivid application is variable selection in regression. Should a model include a particular feature x_j or not? This is a discrete choice: $\mathcal{M}_1$ (model with x_j) versus $\mathcal{M}_0$ (model without x_j).

Under a Gaussian prior on regression coefficients $\beta \sim \mathcal{N}(0, \tau^2 I)$, the marginal likelihood for linear regression is:

$$p(y \mid X, \mathcal{M}) = \mathcal{N}(y;\, 0,\, \tau^2 X X^\top + \sigma^2 I_n)$$

as derived in Chapter 8. For $\mathcal{M}_1$ (with feature x_j) and $\mathcal{M}_0$ (without), the Bayes factor is:

$$\text{BF}_{10} = \frac{p(y \mid X_1)}{p(y \mid X_0)}$$

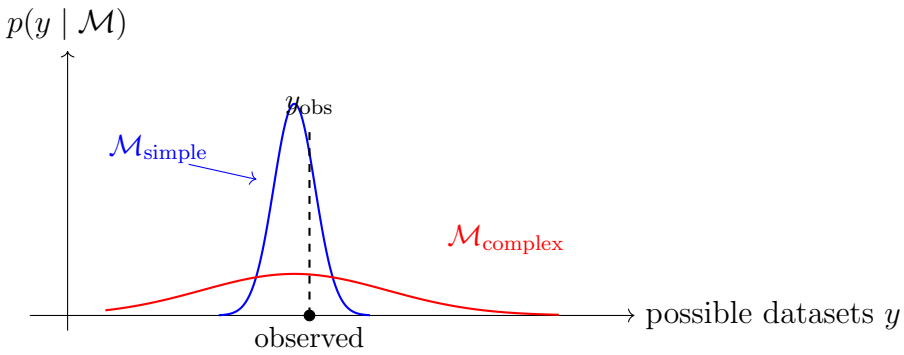
where X_1 includes column j and X_0 does not. If $\text{BF}_{10} > 1$, the feature improves predictive performance enough to justify its inclusion; if $\text{BF}_{10} < 1$, the complexity cost outweighs the fit improvement and the feature should be excluded.

Contrast with p-values. A p-value for the coefficient β_j answers: "If $\beta_j = 0$, how surprising is the observed data?" A Bayes factor answers: "Is the data more probable with $\beta_j \neq 0$ or with $\beta_j = 0$?" These are different questions. A small p-value says the data is surprising under the null; a large Bayes factor says the alternative is

more probable than the null. With large n, even negligibly small effects produce small p-values, because any deviation from zero becomes statistically detectable. The Bayes factor remains calibrated to the actual magnitude of the effect, penalizing models that are more complex than the data requires.

The Automatic Occam's Razor

The most important property of the Bayes factor is that it automatically penalizes complexity. This is not a tuning choice or a regularization parameter; it falls out of the marginalization over parameters.



The simple model $\mathcal{M}_{\text{simple}}$ concentrates its probability over a small set of datasets: it makes sharp predictions. The complex model $\mathcal{M}_{\text{complex}}$ spreads its probability over many possible datasets: it can fit anything, but assigns low probability to any specific dataset. If the observed data y_{obs} is in the region where both models assign similar probability, the simple model wins — because it assigned higher probability to that specific dataset. The Bayes factor rewards models that make sharp, correct predictions.

The Bayes factor is marginal likelihood comparison.

- **Continuous hyperparameters:** maximize the marginal likelihood to select $\hat{\psi}$. Same as empirical Bayes, same cost as a single model fit.
- **Discrete model choice:** compare marginal likelihoods across models. The ratio is the Bayes factor. No held-out data required; automatic complexity penalty built in.
- **Both are the same object:** the marginal likelihood $p(y \mid \mathcal{M}, \psi)$, either optimized over ψ (continuous case) or compared across $\mathcal{M}$ (discrete case).

Limitations of Bayes Factors

Bayes factors are not free of difficulties, and honesty requires acknowledging them.

Prior sensitivity. The marginal likelihood depends on the prior $p(\theta \mid \mathcal{M})$, not just the likelihood. For diffuse priors, the marginal likelihood penalizes complexity heavily (large parameter space, low prior density at any point). For concentrated priors, it penalizes less. Unlike the posterior, where the prior's influence diminishes as n grows, the marginal likelihood can remain sensitive to the prior even for large n when comparing non-nested models.

Improper priors cannot be used. If the prior is improper — does not integrate to one — the marginal likelihood is undefined up to an arbitrary constant, and Bayes factors are meaningless. For model comparison, proper priors are required.

Computational cost for complex models. For models where the marginal likelihood is not analytically available (non-Gaussian likelihoods, deep networks), approximating it requires MCMC, variational inference, or the Laplace approximation. Each introduces additional error on top of the model selection decision.

The Lindley paradox. For a fixed significance level and growing n, a frequentist test will eventually reject the null hypothesis for any nonzero effect. But the Bayes factor can simultaneously favor the null, if the effect is small enough that the data is more probable under the null than under the alternative. This is not a paradox but a genuine difference in what the two frameworks measure: the p-value measures surprise under the null; the Bayes factor measures relative probability of null versus alternative.

BIC: A Cheap Approximation to the Log Marginal Likelihood

For models where the marginal likelihood is expensive or intractable to compute, the **Bayesian Information Criterion** (BIC), introduced by Schwarz (1978), provides a fast approximation. For a model $\mathcal{M}$ with k free parameters and MLE $\hat{\theta}$:

$$\text{BIC}(\mathcal{M}) = -2 \log p(y \mid \hat{\theta}, \mathcal{M}) + k \log n$$

Selecting the model with the lowest BIC is approximately equivalent to selecting the model with the highest marginal likelihood, in the limit of large n. The approximation follows from applying the Laplace approximation to the marginal likelihood integral: expand the log posterior around its mode, integrate the resulting Gaussian, and discard terms that vanish as $n \to \infty$. The $k \log n$ term is the complexity penalty that emerges from this integration — the Bayesian Occam's razor in its crudest form.

BIC is widely used because it requires only the MLE and a parameter count, with no prior specification. But it has real limitations: it is only valid asymptotically, it treats all parameters as equally costly (the penalty is $\log n$ per parameter regardless of the prior), and it provides no posterior probability over models — only a ranking. The Bayes factor is the exact version of what BIC approximates. Practitioners who use BIC are implicitly doing Bayesian model selection with a specific Gaussian prior on the parameters and a large-n approximation, whether or not they know it.

BIC in context.

	Bayes factor	**BIC**
What it is	Exact marginal likelihood ratio	Laplace approximation to log marginal likelihood
Cost	$O(n^3)$ for GP; expensive in general	$O(1)$ given MLE
Prior required	Yes: proper prior on θ	No: implicit Gaussian
Valid for small n	Yes	No: asymptotic
Gives posterior model probabilities	Yes	No: ranking only

Exercise 7.10. Consider a GP with squared exponential kernel applied to $n = 15$ observations. The optimized log marginal likelihoods for three lengthscales are:

ℓ	$\log p(y \mid X, \ell)$
0.1	-31.4
0.5	-22.7
2.0	-27.1

1. Compute the Bayes factor BF between $\ell = 0.5$ and $\ell = 0.1$. What is the posterior probability of $\ell = 0.5$ assuming equal prior probability on the three values?
2. Explain why $\ell = 0.1$ has lower marginal likelihood than $\ell = 0.5$ in terms of the data fit and complexity penalty terms.
3. At $\ell = 0.5$, identify which term in $\log p(y \mid X, \psi) = -\frac{1}{2} y^\top C^{-1} y - \frac{1}{2} \log |C| - \frac{n}{2} \log(2\pi)$ is likely larger than at $\ell = 2.0$, and which is smaller.

Exercise 7.11. In Bayesian linear regression with prior $\beta \sim \mathcal{N}(0, \tau^2 I_p)$ and likelihood $y \sim \mathcal{N}(X\beta, \sigma^2 I_n)$:

1. Show that the marginal likelihood is $p(y \mid X) = \mathcal{N}(y; 0, \tau^2 X X^\top + \sigma^2 I_n)$ by integrating out β.
2. Consider two models: $\mathcal{M}_1$ uses both features (x_1, x_2) and $\mathcal{M}_0$ uses only x_1. With $\tau^2 = 1$, $\sigma^2 = 1$, and a dataset where x_2 is nearly uncorrelated with y, explain qualitatively whether the Bayes factor BF_{10} should be greater or less than 1. What happens to BF_{10} as $\tau^2 \to \infty$?
3. Contrast the Bayes factor with the F-test for including x_2: what does each measure, and when would they disagree?

7.9 Computer Experiments and the Emulator

One of the most practically important and conceptually clean applications of Gaussian processes is the **computer experiment**. The setting is this: a complex physical or engineering system is modeled by a deterministic computer code — a climate simulation, a finite element structural analysis, a computational fluid dynamics solver, a nuclear reactor model. The code takes inputs x (design parameters, initial conditions, material properties) and produces outputs $y = f(x)$ (temperature fields, stress distributions, reaction rates). The function f is deterministic: running the code twice with the same input gives the same output. But it is also a black box: the relationship between inputs and outputs is mediated by millions of lines of code implementing complex physical equations, and there is no analytic formula for $f(x)$.

The problem is that the code is expensive to run. A single evaluation of a high-fidelity climate model may take hours on a supercomputer. A finite element analysis of a jet turbine blade under thermal stress may take days. An engineer who wants to optimize the design — find the input x^* that minimizes stress, maximizes efficiency, or satisfies a safety constraint — cannot afford to evaluate f at the thousands of points that a grid search or gradient-free optimizer would require. They can afford perhaps 50 to 200 evaluations.

This is exactly the black-box optimization problem of Bayesian optimization, but with an additional structure: the function is **deterministic**, not noisy. There is no measurement error. The observation is exact: $y_i = f(x_i)$ precisely.

History: DACE and the Birth of the Emulator

The statistical analysis of computer experiments was formalized by Sacks, Welch, Mitchell, and Wynn in a landmark 1989 paper in *Statistical Science*: "Design and Analysis of Computer Experiments" (DACE). They proposed modeling the unknown function f as a realization of a Gaussian process, fitting the GP to a small number of code evaluations, and using the GP posterior as a **surrogate** or **emulator** for the expensive code. The emulator can be evaluated in microseconds, compared to hours for the original code, and it comes equipped with uncertainty estimates that quantify where the approximation is reliable and where it is not.

The DACE paper was enormously influential. It gave engineers and scientists a principled statistical framework for a problem they had been solving ad hoc for decades, and it connected experimental design, spatial statistics, and Bayesian inference into a coherent whole. The methodology it introduced is now standard in aerospace engineering, automotive design, nuclear engineering, climate science, and drug discovery.

The Deterministic Case: No Noise

When the code is deterministic, there is no observation noise: $\sigma^2 = 0$. The GP must interpolate exactly through the observed points rather than smoothing through them.

The kernel matrix is $K(X, X)$ without the noise term, and the posterior is:

$$f(x_*) \mid f(X) = y \;\sim\; \mathcal{N}(\mu_*, \sigma_*^2)$$

where:

$$\mu_* = k(x_*, X)\, K(X, X)^{-1}\, y$$
$$\sigma_*^2 = k(x_*, x_*) - k(x_*, X)\, K(X, X)^{-1}\, k(X, x_*)$$

Notice that $\sigma_*^2 = 0$ whenever x_* is one of the training points x_i: the emulator is certain at observed locations, because the code is deterministic. The uncertainty grows as x_* moves away from the observed points, reflecting the fact that we have not run the code there.

This is the one-million-people picture applied to functions. The GP prior is the prior over candidate functions. Running the code at $x_1, \ldots, x_n$ filters out all candidate functions that do not pass through $(x_i, f(x_i))$ exactly. The posterior is the distribution over the remaining candidates. Evaluating the emulator at a new x_* is asking: among all functions consistent with the code evaluations so far, what is the distribution of $f(x_*)$?

A Concrete Example: Modeling a Wing's Lift Coefficient

We illustrate with a simplified aerodynamics example. An aerospace engineer is designing a wing and wants to understand how the **lift coefficient** C_L depends on two design parameters:

- $x_1 \in [0, 15]$: angle of attack (degrees)
- $x_2 \in [0.05, 0.20]$: wing thickness-to-chord ratio

Computing $C_L(x_1, x_2)$ requires running a computational fluid dynamics (CFD) solver, which takes 3 hours per evaluation on the available cluster. The engineer has a budget of $n = 20$ evaluations.

Step 1: Design the experiment. The engineer must choose where to run the code. A naive grid over a 5×4 mesh of (x_1, x_2) values would use all 20 runs but cluster them in a regular pattern that may miss important features. The standard approach for computer experiments is a **Latin hypercube design**: partition each input dimension into n equal intervals and place exactly one design point in each interval for each dimension. This ensures the runs are spread evenly across the input space without clustering.

The 20 Latin hypercube design points are run through the CFD solver over the course of one week, producing a dataset:

$$\mathcal{D} = \{(x_i, C_L(x_i))\}_{i=1}^{20}$$

Step 2: Fit the GP emulator. Fit a GP with a squared exponential kernel:

$$k(x, x') = \sigma_f^2 \exp\left(-\frac{(x_1 - x_1')^2}{2\ell_1^2} - \frac{(x_2 - x_2')^2}{2\ell_2^2} \right)$$

with separate lengthscales ℓ_1 and ℓ_2 per dimension (ARD: Automatic Relevance Determination). The hyperparameters $(\ell_1, \ell_2, \sigma_f^2)$ are optimized by maximizing the log marginal likelihood:

$$\log p(C_L \mid X, \psi) = -\frac{1}{2} C_L^\top K^{-1} C_L - \frac{1}{2} \log |K| - \frac{n}{2} \log(2\pi)$$

where $K = K(X, X)$ has no noise term (deterministic code). Suppose the optimization yields $\hat{\ell}_1 = 4.2$ and $\hat{\ell}_2 = 0.08$: the lift coefficient varies more slowly with thickness than with angle of attack, which makes physical sense.

Step 3: Interrogate the emulator. Once fitted, the emulator can be evaluated at any (x_1, x_2) in milliseconds. The engineer can:

- **Predict with uncertainty.** At any untested design point (x_1^*, x_2^*), the emulator provides a posterior mean μ_* (best estimate of C_L) and posterior standard deviation σ_* (uncertainty in that estimate). The 95% credible interval $[\mu_* - 1.96\sigma_*, \mu_* + 1.96\sigma_*]$ tells the engineer how much to trust the prediction.

- **Optimize.** Find the design (x_1^*, x_2^*) that maximizes C_L. This is Bayesian optimization: use the emulator posterior and expected improvement to select the next CFD run efficiently. Starting from the 20 initial runs, perhaps 5–10 additional targeted runs are enough to locate the optimum to engineering precision.

- **Sensitivity analysis.** Compute $\partial \mu_* / \partial x_1$ and $\partial \mu_* / \partial x_2$ from the emulator's posterior mean: which input dimension most strongly influences the output? The ARD lengthscales $\hat{\ell}_1$ and $\hat{\ell}_2$ give an immediate answer: smaller ℓ_j means more variation in direction j, hence greater sensitivity.

- **Uncertainty propagation.** If the input x is itself uncertain — manufacturing tolerances make x_2 known only to ± 0.005 — propagate this uncertainty through the emulator to obtain a distribution over C_L. This requires integrating the emulator over the input uncertainty, which is tractable because the emulator is a cheap analytic function.

Step 4: Validate. Before trusting the emulator, validate it with a small number of held-out code runs. Typical practice is to hold out 5–10 of the 20 design points, fit the emulator on the rest, and check whether the held-out values fall within the emulator's credible intervals. If 95% of the held-out values fall within the 95% credible intervals, the emulator is well-calibrated and the hyperparameters are appropriate.

Why the Deterministic GP Works: Interpolation as Posterior Conditioning

The deterministic GP emulator works for a reason that is invisible in the frequentist framework: it produces a posterior that is consistent with everything we know. Each code evaluation is exact information — $f(x_i)$ is known precisely — and the posterior conditions on this information exactly, producing zero uncertainty at observed points and growing uncertainty away from them. This is the correct behavior. A method that produces nonzero uncertainty at observed points (as a noisy GP would) is stating uncertainty about something that is not uncertain: the code is deterministic.

The uncertainty in the emulator is **epistemic**: it arises from not having run the code at untested points, not from any randomness in the code itself. As more code runs are added, the epistemic uncertainty shrinks. If we could run the code everywhere, the uncertainty would vanish everywhere. The GP posterior quantifies exactly this: how much we do not know, and where.

The emulator in one sentence. A GP fitted to a small set of expensive code evaluations produces a cheap, accurate, uncertainty-aware surrogate for the code: it interpolates exactly through observed outputs, predicts at untested inputs with calibrated uncertainty, and can be used for optimization, sensitivity analysis, and uncertainty propagation at a tiny fraction of the cost of running the original code.

Extensions and Current Practice

The basic DACE framework has been extended in many directions that are active research areas.

Multi-fidelity emulation. Many engineering problems have codes of multiple fidelity levels: a fast, approximate solver and a slow, accurate one. Multi-fidelity GPs combine runs from both, using the cheap code to provide many evaluations and the expensive code to correct the cheap code's errors. The result is a more accurate emulator than either code alone could provide for the same computational budget.

High-dimensional inputs. When the input space has $p = 100$ or more dimensions — as in climate models with many tunable parameters — the ARD kernel has p lengthscales to estimate, and the Latin hypercube design must cover a very high-dimensional space. Active subspace methods and dimension reduction techniques are combined with GPs to handle these cases.

Functional outputs. Many codes produce not a scalar but a field: a temperature distribution, a pressure profile, a time series. Output-space dimension reduction (e.g., principal component analysis of the output field) is combined with a separate GP emulator for each principal component.

Current applications. GP emulators are standard tools in:

- **Climate science:** emulating global climate models to explore the sensitivity of climate projections to uncertain physical parameters (cloud formation rates,

aerosol properties). The UK Met Office and climate modelling groups worldwide use GP emulators routinely.

- **Nuclear engineering:** emulating reactor safety codes to quantify the probability of core damage under various accident scenarios, without running thousands of expensive simulations.
- **Automotive and aerospace:** emulating crash simulations and aerodynamics codes for design optimization and uncertainty quantification.
- **Systems biology:** emulating differential equation models of gene regulatory networks, protein folding, or pharmacokinetics to fit parameters from experimental data.

In all these fields, the GP emulator plays the same role: it is the statistical model that connects the expensive simulation to the scientific or engineering question, providing predictions and uncertainties where the simulation cannot be run.

Exercise 7.12. A structural engineer has a finite element code that computes the maximum deflection $y = f(x_1, x_2)$ of a bridge deck under load, where $x_1 \in [10, 30]$ m is the span length and $x_2 \in [0.5, 2.0]$ m is the deck thickness. Five code runs produce:

x_1 (m)	x_2 (m)	y (mm)
10	0.5	12.3
10	2.0	3.1
20	1.0	18.7
30	0.5	52.4
30	2.0	13.2

1. With $\sigma^2 = 0$ (deterministic code), verify that the GP posterior mean passes exactly through all five observed points: $\mu_*(x_i) = y_i$ for each i.
2. The engineer needs to predict deflection at $(x_1, x_2) = (20, 1.5)$. Using the SE kernel with $\ell_1 = 10$, $\ell_2 = 0.75$, $\sigma_f^2 = 400$, compute the posterior mean μ_* and standard deviation σ_* at this point.
3. The safety requirement is that deflection must be below 20 mm with 99% probability. Compute $P(f(20, 1.5) < 20\,\text{mm} \mid \mathcal{D})$ from the GP posterior and determine whether the safety requirement is met.
4. Explain why the posterior standard deviation at $(20, 1.5)$ is larger than at $(20, 1.0)$, even though both points are the same distance from the training data in x_1.

Exercise 7.13. A climate scientist has a model with three uncertain parameters $x = (x_1, x_2, x_3)$: cloud feedback, aerosol forcing, and ocean heat uptake. They run the

model at $n = 30$ Latin hypercube design points and fit a GP emulator with ARD kernel. The optimized lengthscales are $\hat{\ell}_1 = 0.12$, $\hat{\ell}_2 = 0.87$, $\hat{\ell}_3 = 0.43$ (inputs scaled to $[0, 1]$).

1. Which parameter has the strongest influence on the model output? Which has the weakest? Explain how the lengthscales encode this information.

2. The scientist wants to reduce uncertainty in the output by constraining one parameter with an additional 10 experiments. Based on the lengthscales alone, which parameter should they prioritize? Justify your answer.

3. Explain why the log marginal likelihood is a better criterion than cross-validation for selecting the lengthscales in this computer experiment setting, where the code is deterministic and $n = 30$ is small.

7.10 Uncertainty Quantification

Uncertainty quantification (UQ) is the science of characterizing, propagating, and reducing uncertainty in computational and mathematical models of physical systems. It asks: given that our model inputs are uncertain, our model structure is approximate, and our observations are noisy, how uncertain should we be about our model's predictions? And which sources of uncertainty matter most?

UQ has become a central concern across science and engineering. A climate model prediction of global temperature in 2100 is only useful if accompanied by a credible range. A nuclear reactor safety analysis is only convincing if it quantifies the probability of failure, not just whether failure is possible. A drug dosing recommendation is only responsible if it accounts for uncertainty in the pharmacokinetic model. In each case, a point prediction without uncertainty is incomplete at best and dangerous at worst.

The Gaussian process is the natural tool for UQ because it provides exactly what UQ requires: a probability distribution over possible outputs, not just a point estimate.

The Two Sources of Uncertainty

UQ distinguishes two fundamentally different types of uncertainty, and treating them separately is important.

Aleatoric uncertainty (from the Latin *alea*, die) is irreducible randomness: uncertainty that cannot be reduced by collecting more data or building a better model. Measurement noise is aleatoric: even with a perfect instrument, quantum fluctuations impose a floor on precision. Weather is aleatoric beyond a few weeks: the atmosphere is chaotic and no model, however detailed, can predict its state arbitrarily far in advance. Aleatoric uncertainty is a property of the world, not of our knowledge.

Epistemic uncertainty (from the Greek *episteme*, knowledge) is reducible uncertainty: uncertainty that arises from limited knowledge and could in principle be

reduced by more data, better models, or more computation. Not knowing the true value of a physical constant is epistemic: it is fixed, and more measurements will narrow our uncertainty. Not having run a simulation at a particular input is epistemic: running it there would resolve the uncertainty completely. Epistemic uncertainty is a property of the observer, not the world.

Aleatoric vs. epistemic uncertainty.

	Aleatoric	**Epistemic**
Source	World is random	Observer lacks knowledge
Reducible?	No	Yes
Examples	Measurement noise, quantum effects, weather	Unknown parameters, untested inputs, model error
In GP regression	σ^2: noise variance	$\sigma_*^2 - \sigma^2$: parameter and interpolation uncertainty
Bayesian treatment	Modeled by likelihood	Modeled by posterior

The GP posterior predictive variance $\sigma_*^2 = \sigma^2 + \sigma^2 x_*^\top (X^\top X + \lambda I)^{-1} x_*$ decomposes exactly into these two components: aleatoric noise σ^2 and epistemic parameter uncertainty $\sigma^2 x_*^\top (X^\top X + \lambda I)^{-1} x_*$. Only the epistemic component shrinks as more data arrives.

The Four Tasks of UQ

UQ encompasses four related but distinct tasks. The GP provides a natural solution to each.

Task 1: Forward uncertainty propagation. Given uncertain inputs $x \sim p(x)$, what is the distribution of the output $y = f(x)$?

If f is an expensive code, we cannot afford to sample x from $p(x)$ thousands of times and run the code each time (Monte Carlo). Instead: fit a GP emulator $\hat{f}$ to a small number of code runs, then sample $x_1, \ldots, x_S \sim p(x)$ and evaluate the cheap emulator:

$$p(y) \approx \frac{1}{S} \sum_{s=1}^{S} \delta(\hat{f}(x^{(s)}))$$

The emulator makes this feasible: $S = 100{,}000$ samples from the emulator costs milliseconds; $S = 100{,}000$ samples from the code would cost years. The emulator

uncertainty also propagates: the output uncertainty has contributions from input uncertainty and from the emulator's own uncertainty about f.

Task 2: Sensitivity analysis. Which inputs x_j most strongly influence the output? Knowing this tells engineers where to focus measurement effort, which parameters require tight manufacturing tolerances, and which assumptions in the model matter most.

The GP provides two complementary answers:

Local sensitivity: the posterior mean gradient $\partial\mu_*(x)/\partial x_j$ at a nominal design point. This measures how much the predicted output changes per unit change in input j around the current design.

Global sensitivity (Sobol indices): the fraction of the total output variance attributable to each input, integrated over the input distribution:

$$S_j = \frac{\mathrm{Var}_{x_j}(\mathbb{E}_{x_{-j}}[f(x) \mid x_j])}{\mathrm{Var}(f(x))}$$

Computing Sobol indices by Monte Carlo requires thousands of code evaluations. With a GP emulator, the same integrals can be computed analytically (for Gaussian input distributions) or at negligible cost by sampling from the emulator. The ARD lengthscales $\hat{\ell}_j$ provide a fast proxy: small ℓ_j indicates high sensitivity in direction j.

Task 3: Calibration. Given observations of the real system y_{obs} and a computer model $f(x, \theta)$ with unknown parameters θ (not design inputs but model parameters — physical constants, material properties, closure terms in differential equations), find the values of θ that make the model consistent with the observations.

This is the inverse problem: infer θ from $y_{\text{obs}} = f(x, \theta) + \varepsilon$. Kennedy and O'Hagan (2001) proposed the influential framework:

$$y_{\text{obs}}(x) = f(x, \theta) + \delta(x) + \varepsilon$$

where $\delta(x)$ is a **model discrepancy** GP: a Gaussian process that accounts for the systematic difference between the best computer model and reality, even at the true θ. The model is never perfect; $\delta(x)$ captures what it gets wrong. The full Bayesian treatment places priors on θ and δ, and infers both simultaneously from the observations.

The Kennedy-O'Hagan framework.

$$y_{\text{obs}}(x) = \underbrace{f(x, \theta)}_{\text{computer model}} + \underbrace{\delta(x)}_{\text{model discrepancy (GP)}} + \underbrace{\varepsilon}_{\text{noise}}$$

Not accounting for $\delta(x)$ — assuming the model is perfect at the true θ — leads to biased parameter estimates: the inferred θ compensates for model error by moving away from the physically correct value. The model discrepancy GP is the honest acknowledgment that all models are wrong, and Bayes rule propagates this honesty

through to the parameter estimates.

Task 4: Decision making under uncertainty. Given uncertain model predictions, what decision should be made? A bridge designer who knows the deflection is 18 ± 4 mm (mean $\pm$ standard deviation) against a safety limit of 20 mm must decide whether to approve the design. A drug regulator who knows the efficacy is $65 \pm 12\%$ against a minimum threshold of 60% must decide whether to approve the drug.

The GP posterior predictive provides the full distribution needed for this decision:

$$P(f(x^*) > \text{threshold} \mid \mathcal{D}) = 1 - \Phi\left(\frac{\text{threshold} - \mu_*}{\sigma_*}\right)$$

This is the probability that the true output exceeds the safety or efficacy threshold, given the data. It is the answer to the actual question the decision maker is asking, and it is only available because the GP provides a full distribution rather than a point estimate.

A Concrete Example: Climate Sensitivity UQ

Equilibrium climate sensitivity (ECS) is the long-run global temperature increase resulting from a doubling of atmospheric CO_2. It is one of the most consequential uncertain quantities in science: if ECS is 2°C, climate change is manageable with moderate mitigation; if it is 5°C, it is catastrophic.

ECS cannot be measured directly. It must be inferred from a combination of:

- Paleoclimate data (ice cores, sediment records) showing temperature and CO_2 over millions of years.
- Instrumental records of recent temperature and forcing.
- Physical understanding of the climate system, encoded in general circulation models (GCMs).

Each GCM has uncertain parameters θ (cloud microphysics, convection parameterizations, ocean mixing) that affect ECS. Running a GCM to equilibrium takes months of supercomputer time. UQ proceeds as follows:

Step 1: Design. Run the GCM at $n = 50$ parameter settings chosen by Latin hypercube design. Record ECS at each setting.

Step 2: Emulate. Fit a GP emulator to the 50 (parameter setting, ECS) pairs. The emulator predicts ECS at any parameter setting in milliseconds, with calibrated uncertainty.

Step 3: Calibrate. Use the observed paleoclimate and instrumental data to update the prior over θ to a posterior, using the Kennedy-O'Hagan framework. The model discrepancy GP accounts for the fact that even the best GCM is not a perfect model of the real climate system.

Step 4: Propagate. Push the posterior over θ through the emulator to obtain a posterior distribution over ECS:

$$p(\text{ECS} \mid \text{observations}) = \int p(\text{ECS} \mid \theta)\, p(\theta \mid \text{observations})\, d\theta$$

The result is a probability distribution over ECS, not a point estimate. The IPCC Sixth Assessment Report (2021) reports ECS as "likely between 2.5°C and 4.0°C" with a best estimate of 3.0°C. This range is the output of exactly this kind of UQ analysis, combining multiple lines of evidence through Bayesian inference.

The GP emulator is what makes this computationally feasible: the posterior integration over θ requires millions of evaluations of the climate model, which would take thousands of years on available hardware. With the emulator, it takes hours.

UQ and the Honest Model

There is a deeper point beneath the technical machinery. UQ forces intellectual honesty about what a model can and cannot tell us. A model without uncertainty quantification implicitly claims to know more than it does. A point prediction of global temperature in 2100 — or bridge deflection, or drug efficacy — presented without uncertainty conceals the limitations of the model from the decision maker.

The GP posterior is an honest model. It says: here is what I know (the posterior mean, informed by observations), and here is what I do not know (the posterior variance, encoding uncertainty away from observed data). As more observations arrive, the uncertainty shrinks appropriately. When the model is asked to extrapolate beyond its training data, the uncertainty grows appropriately. The model is not pretending to know what it does not know.

This honesty has a Bayesian pedigree. The prior encodes what was known before observations. The likelihood encodes what the observations say. The posterior encodes what is known after. The posterior predictive propagates this knowledge — and this uncertainty — to new predictions. Each step is transparent, each source of uncertainty is tracked, and the final prediction comes with a probability distribution that can be used directly for decisions. This is what UQ, at its best, looks like. And the GP is the tool that makes it computationally tractable for the problems that matter most.

Exercise 7.14. A pharmacokineticist models the blood concentration $C(t)$ of a drug using a two-compartment model with uncertain parameters $\theta = (k_a, k_e, V_d)$ (absorption rate, elimination rate, volume of distribution). The model is a system of ODEs solved numerically, taking 0.5 seconds per evaluation. Patient measurements are available at $n = 8$ time points.

1. The pharmacokineticist wants to compute the probability that the concentration exceeds a toxic threshold of 5 μg/mL at $t = 2$ hours, integrating over uncertainty in θ. Explain why a GP emulator for $C(2; \theta)$ as a function of θ is preferable to direct Monte Carlo sampling from $p(\theta \mid \text{data})$.

2. The three parameters have prior distributions $k_a \sim \text{Gamma}(2, 1)$, $k_e \sim \text{Gamma}(3, 2)$, $V_d \sim \mathcal{N}(50, 100)$. How would you design the $n = 30$ training runs for the emulator, and what design criterion would you use?

3. After fitting the emulator, the optimized ARD lengthscales are $\hat{\ell}_{k_a} = 0.3$, $\hat{\ell}_{k_e} = 1.8$, $\hat{\ell}_{V_d} = 0.9$ (inputs standardized). Which parameter most strongly influences $C(2; \theta)$? What does this imply for future experiments aimed at reducing uncertainty in the toxic threshold probability?

Exercise 7.15. Consider the climate sensitivity example. Suppose the GP emulator for ECS as a function of the uncertain parameter vector $\theta \in [0, 1]^5$ has been fitted with a squared exponential ARD kernel. The posterior over θ given observational data is approximately $\mathcal{N}(\hat{\theta}, \Sigma_\theta)$.

1. Explain why the posterior distribution over ECS, $p(\text{ECS} \mid \text{observations})$, is not Gaussian even though $p(\theta \mid \text{observations})$ is approximately Gaussian and the emulator posterior mean is a smooth function of θ.

2. The model discrepancy term $\delta(x)$ in the Kennedy-O'Hagan framework is modeled as a GP. Explain what would happen to the inferred $\hat{\theta}$ if $\delta(x)$ were omitted: in which direction would the parameter estimates be biased, and why?

3. The IPCC reports a "likely range" of 2.5°C to 4.0°C for ECS. In Bayesian terms, what probability is associated with "likely"? What prior model probabilities and likelihood assumptions would you need to reproduce this range from a GP emulator analysis?

Chapter 8

Gibbs Sampler

The preceding chapters established what Bayesian inference *is*: prior times likelihood, normalized to a posterior. For conjugate models — Beta-Binomial, Gaussian-Gaussian — the normalizing constant is available in closed form and the posterior is a standard distribution. For everything else, the posterior is known only up to a constant, and that constant requires integrating over the entire parameter space. In most real problems this integral is intractable.

Markov chain Monte Carlo (MCMC) methods sidestep the normalization problem entirely. Instead of computing the posterior analytically, they construct a random walk through the parameter space whose long-run distribution is the posterior. After the walk has run long enough, the visited points are approximate samples from the posterior, and we can estimate any quantity of interest — means, variances, credible intervals — by averaging over those samples. This chapter develops the most intuitive MCMC algorithm: Gibbs sampling.

8.1 One Million People on an Island

We begin with a picture that makes Gibbs sampling obvious.

A Uniform Distribution: Shuffling on a Grid

Imagine one million people standing on an island of irregular shape. The island is two-dimensional: each person has a position (x, y). Suppose initially the million people are distributed uniformly over the island — every location equally likely.

Now perform the following operation. Divide the island into a fine grid of horizontal strips (rows). Within each horizontal strip, redistribute all the people uniformly along that strip. Then divide the island into vertical strips (columns) and redistribute all the people uniformly within each vertical strip.

Does this change the uniform distribution? No. A uniform distribution within every row and within every column is exactly the uniform distribution over the whole

island. The row-wise and column-wise redistributions leave the joint uniform distribution invariant.

Now start from a degenerate initial condition: all one million people stacked at a single point. After one round of row-then-column redistribution, the people are no longer concentrated at a point. After many rounds, they disperse. The long-run distribution — the distribution the population converges to under repeated row-then-column shuffling — is the uniform distribution over the island. The island's shape determines where people can be; the shuffling procedure finds that shape automatically.

A General Distribution: People Under a Surface

Now generalize. Instead of a flat island, imagine that the people can live stacked on top of each other, with the local density of people at location (x, y) determined by a surface $p(x, y)$ above that point. Where the surface is tall, people are densely packed; where it is low, they are sparse. This is exactly the probability density picture from Chapter 1: $p(x, y)\,\Delta x\,\Delta y$ is the fraction of people in the small rectangle near (x, y).

The shuffling logic still applies, with one modification: instead of redistributing uniformly within each strip, we redistribute according to the *conditional* distribution. Within the horizontal strip at height y, the density of people along the x-axis is $p(x \mid y)$ — the conditional of x given y. Within the vertical strip at position x, the density along the y-axis is $p(y \mid x)$.

The Gibbs sampling picture. Gibbs sampling is the row-then-column shuffling procedure for a general distribution $p(x, y)$:

1. Given the current y-position of each person, redraw their x-position from $p(x \mid y)$.
2. Given the new x-position, redraw their y-position from $p(y \mid x)$.
3. Repeat.

If all one million people start at the same point, repeated shuffling disperses them until their distribution matches $p(x, y)$. Each shuffle leaves $p(x, y)$ invariant: if the population is currently distributed according to $p(x, y)$, it remains so after a shuffle. The joint distribution is the fixed point of the shuffling procedure.

Why does the shuffle leave $p(x, y)$ invariant? After the x-shuffle: each person at position y draws a new x from $p(x \mid y)$. The fraction arriving at (x, y) is $p(x \mid y) \cdot p(y) = p(x, y)$. The joint is unchanged. The same argument applies to the y-shuffle. One full round of row-then-column shuffling is a fixed point of the joint distribution.

From Picture to Algorithm

In practice, we track a single person rather than one million. That person's position $(x^{(t)}, y^{(t)})$ at step t is one sample. As t grows, the sequence of positions $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots$ traces out a Markov chain whose stationary distribution is $p(x, y)$. After a **burn-in** period during which the chain converges to stationarity, each subsequent position is approximately a draw from $p(x, y)$.

Gibbs Sampling Algorithm. To sample from $p(\theta_1, \ldots, \theta_d \mid \mathcal{D})$, initialize $\theta^{(0)}$ arbitrarily and repeat for $t = 1, 2, \ldots$:

$$\begin{aligned}
\theta_1^{(t)} &\sim p\Big(\theta_1 \mid \theta_2^{(t-1)}, \ldots, \theta_d^{(t-1)}, \mathcal{D}\Big) \\
\theta_2^{(t)} &\sim p\Big(\theta_2 \mid \theta_1^{(t)}, \theta_3^{(t-1)}, \ldots, \theta_d^{(t-1)}, \mathcal{D}\Big) \\
&\vdots \\
\theta_d^{(t)} &\sim p\Big(\theta_d \mid \theta_1^{(t)}, \ldots, \theta_{d-1}^{(t)}, \mathcal{D}\Big)
\end{aligned}$$

Each one-dimensional conditional $p(\theta_j \mid \theta_{-j}, \mathcal{D})$ is a slice through the joint posterior, often tractable even when the joint is not.

Remark 8.1 (Mixing and correlation)**.** When the target distribution $p(x, y)$ is concentrated near a diagonal ridge — high correlation between x and y — the row-then-column shuffling makes small steps along the ridge. A person can only move horizontally (adjust x) or vertically (adjust y) at each step, but the high-density ridge runs diagonally. Progress along the ridge is therefore slow: many small zigzag steps are needed to traverse it. This is **slow mixing**: the chain is correct in the limit, but converges slowly. Slow mixing is the central practical limitation of Gibbs sampling and is most severe when the posterior has strong correlations.

8.2 A Two-Component Gaussian Mixture Model

We now apply Gibbs sampling to a problem where it is most naturally at home: the mixture model. We restrict to two components so that all computations reduce to the Beta-Binomial and Gaussian-Gaussian posteriors we derived in Chapter 4 — no new mathematics is required, only new combinations.

The Model

We observe n scalar data points $x_1, \ldots, x_n \in \mathbb{R}$. We believe they come from a mixture of two Gaussian components, but we do not know which component generated each point. The model is:

$$
\begin{aligned}
\pi &\sim \text{Beta}(\alpha_0, \alpha_0) && \text{(mixing weight, symmetric prior)} \\
\mu_1 &\sim \mathcal{N}(m, s^2) && \text{(mean of component 1)} \\
\mu_2 &\sim \mathcal{N}(m, s^2) && \text{(mean of component 2)} \\
z_i \mid \pi &\sim \text{Bernoulli}(\pi), \quad i = 1, \ldots, n && \text{(latent assignment: 0 or 1)} \\
x_i \mid z_i, \mu_1, \mu_2 &\sim \mathcal{N}(\mu_{z_i}, \sigma^2) && \text{(observation from assigned component)}
\end{aligned}
$$

The parameters are:

- $\pi \in [0, 1]$: the probability that any observation comes from component 1.
- $\mu_1, \mu_2 \in \mathbb{R}$: the means of the two Gaussian components.
- $\sigma^2 > 0$: the common variance, treated as known.

The latent variables are the **component assignments** $z_i \in \{0, 1\}$: $z_i = 1$ means observation i came from component 1, and $z_i = 0$ means it came from component 2. We do not observe the z_i directly.

The hyperparameters α_0, m, s^2 are fixed in advance. A symmetric prior $\alpha_0 = 1$ gives $\text{Beta}(1, 1) = \text{Uniform}[0, 1]$.

The **joint distribution** over all unknowns and observations is:

$$
p(\pi, \mu_1, \mu_2, z, x) = p(\pi)\, p(\mu_1)\, p(\mu_2) \prod_{i=1}^{n} p(z_i \mid \pi)\, p(x_i \mid z_i, \mu_1, \mu_2)
$$

where $z = (z_1, \ldots, z_n)$ and $x = (x_1, \ldots, x_n)$.

The joint posterior we want to sample from is:

$$
p(\pi, \mu_1, \mu_2, z \mid x) \propto p(\pi)\, p(\mu_1)\, p(\mu_2) \prod_{i=1}^{n} p(z_i \mid \pi)\, p(x_i \mid z_i, \mu_1, \mu_2)
$$

The normalizing constant requires summing over all 2^n possible assignment vectors z and integrating over (π, μ_1, μ_2): intractable for any realistic n. But each of the three conditionals is tractable, as we now derive.

Conditional 1: The Assignment Variables z_i

We derive $p(z_i \mid \pi, \mu_1, \mu_2, x_i)$, the conditional distribution of each assignment given everything else. Since the z_i are conditionally independent given (π, μ_1, μ_2, x), we can update each one separately.

By Bayes rule:

$$
\begin{aligned}
p(z_i = 1 \mid \pi, \mu_1, \mu_2, x_i) &\propto p(z_i = 1 \mid \pi)\, p(x_i \mid z_i = 1, \mu_1) \\
&= \pi \cdot \mathcal{N}(x_i; \mu_1, \sigma^2)
\end{aligned}
$$

$$\begin{aligned} p(z_i = 0 \mid \pi, \mu_1, \mu_2, x_i) &\propto p(z_i = 0 \mid \pi)\, p(x_i \mid z_i = 0, \mu_2) \\ &= (1 - \pi) \cdot \mathcal{N}(x_i; \mu_2, \sigma^2) \end{aligned}$$

Normalizing:

$$p(z_i = 1 \mid \pi, \mu_1, \mu_2, x_i) = \frac{\pi\, \mathcal{N}(x_i; \mu_1, \sigma^2)}{\pi\, \mathcal{N}(x_i; \mu_1, \sigma^2) + (1 - \pi)\, \mathcal{N}(x_i; \mu_2, \sigma^2)} \;=:\; r_i$$

The quantity r_i is the **responsibility** of component 1 for observation i: the posterior probability that x_i was generated by component 1. It is proportional to the prior weight π times the likelihood of x_i under component 1. This is exactly the one million people picture: among all people who end up at value x_i, what fraction came from component 1?

The conditional distribution is Bernoulli:

$$z_i \mid \pi, \mu_1, \mu_2, x_i \;\sim\; \text{Bernoulli}(r_i)$$

Intuition. If x_i is close to μ_1 and far from μ_2, then $\mathcal{N}(x_i; \mu_1, \sigma^2) \gg \mathcal{N}(x_i; \mu_2, \sigma^2)$, and $r_i \approx 1$: the observation is almost certainly from component 1. If x_i falls between the two means, $r_i \approx \pi$: uncertainty is high and the prior weight determines the assignment.

Conditional 2: The Mixing Weight π

Given the assignments $z = (z_1, \ldots, z_n)$, let:

$$n_1 = \sum_{i=1}^{n} z_i \quad \text{(number assigned to component 1)}$$

$$n_0 = n - n_1 \quad \text{(number assigned to component 2)}$$

The conditional distribution of π given z is:

$$\begin{aligned} p(\pi \mid z) &\propto p(\pi) \prod_{i=1}^{n} p(z_i \mid \pi) \\ &\propto \pi^{\alpha_0 - 1}(1 - \pi)^{\alpha_0 - 1} \cdot \pi^{n_1}(1 - \pi)^{n_0} \\ &= \pi^{\alpha_0 + n_1 - 1}(1 - \pi)^{\alpha_0 + n_0 - 1} \end{aligned}$$

$$\pi \mid z \;\sim\; \text{Beta}(\alpha_0 + n_1,\; \alpha_0 + n_0)$$

This is the Beta-Binomial conjugate update from Chapter 4, applied to the latent counts. The prior $\text{Beta}(\alpha_0, \alpha_0)$ contributes α_0 pseudo-counts to each component; the data adds the actual counts n_1 and n_0. The posterior mean is:

$$\mathbb{E}[\pi \mid z] = \frac{\alpha_0 + n_1}{2\alpha_0 + n}$$

When n is large, this approaches n_1/n — the fraction of observations currently assigned to component 1.

Conditional 3: The Component Means μ_1, μ_2

Given the assignments z, the data splits into two groups:

$$\mathcal{D}_1 = \{x_i : z_i = 1\}, \quad \mathcal{D}_0 = \{x_i : z_i = 0\}$$

Let $n_k = |\mathcal{D}_k|$ and $\bar{x}_k = \frac{1}{n_k}\sum_{i:z_i=k} x_i$ be the sample mean of group k. The means μ_1 and μ_2 are conditionally independent given z, so we update each separately.

For μ_1: the likelihood from $\mathcal{D}_1$ is $\prod_{i:z_i=1}\mathcal{N}(x_i; \mu_1, \sigma^2)$, and the prior is $\mu_1 \sim \mathcal{N}(m, s^2)$. This is exactly the Gaussian-Gaussian conjugate update from Chapter 4:

Posterior for μ_1:

$$\mu_1 \mid z, x \;\sim\; \mathcal{N}(\mu_1^*, \tau_1^2)$$

where:

$$\frac{1}{\tau_1^2} = \frac{1}{s^2} + \frac{n_1}{\sigma^2}, \qquad \mu_1^* = \tau_1^2\Big(\frac{m}{s^2} + \frac{n_1\bar{x}_1}{\sigma^2}\Big)$$

The posterior mean is the precision-weighted average of the prior mean m and the group sample mean $\bar{x}_1$, exactly as in the speed of light example. Similarly for μ_2: replace subscript 1 with 0 throughout.

Intuition. Given which observations belong to component 1, estimating μ_1 is just the standard Gaussian mean estimation problem with n_1 data points. The Gibbs sampler exploits this: the complex joint posterior over (π, μ_1, μ_2, z) decomposes, conditional on the assignments, into three simple problems we already know how to solve.

The Gibbs Sampler for the Mixture Model

Putting the three conditionals together:

Gibbs Sampler for Two-Component Gaussian Mixture:

Initialize: set $\mu_1^{(0)}, \mu_2^{(0)}, \pi^{(0)}$ to reasonable starting values (e.g., $\mu_1^{(0)} = \bar{x} + 1$, $\mu_2^{(0)} = \bar{x} - 1$, $\pi^{(0)} = 0.5$).

For $t = 1, 2, \ldots, T$:

1. **Update assignments.** For each $i = 1, \ldots, n$: compute responsibility

$$r_i^{(t)} = \frac{\pi^{(t-1)}\,\mathcal{N}(x_i; \mu_1^{(t-1)}, \sigma^2)}{\pi^{(t-1)}\,\mathcal{N}(x_i; \mu_1^{(t-1)}, \sigma^2) + (1-\pi^{(t-1)})\,\mathcal{N}(x_i; \mu_2^{(t-1)}, \sigma^2)}$$

   and draw $z_i^{(t)} \sim \text{Bernoulli}(r_i^{(t)})$.

2. **Update mixing weight.** Compute $n_1^{(t)} = \sum_i z_i^{(t)}$, $n_0^{(t)} = n - n_1^{(t)}$, and draw:

$$\pi^{(t)} \sim \text{Beta}(\alpha_0 + n_1^{(t)},\ \alpha_0 + n_0^{(t)})$$

3. **Update component means.** Compute group sample means $\bar{x}_1^{(t)}$ and $\bar{x}_0^{(t)}$, and draw:

$$\mu_1^{(t)} \sim \mathcal{N}\left(\mu_1^{*(t)},\ \tau_1^{2(t)}\right), \quad \mu_2^{(t)} \sim \mathcal{N}\left(\mu_2^{*(t)},\ \tau_0^{2(t)}\right)$$

   using the precision-weighted formulas above. If $n_1^{(t)} = 0$ (no observations assigned to component 1), draw $\mu_1^{(t)}$ from the prior $\mathcal{N}(m, s^2)$ directly.

After T_{burn} burn-in iterations, collect samples $\{(\pi^{(t)}, \mu_1^{(t)}, \mu_2^{(t)}, z^{(t)})\}_{t=T_{\text{burn}}+1}^{T}$ as approximate draws from the joint posterior.

A Numerical Example

We run the Gibbs sampler on a small dataset to trace through the calculations explicitly.

Data: $n = 8$ observations:

$$x = (-2.1,\ -1.8,\ -1.5,\ -0.3,\ 0.8,\ 1.2,\ 1.9,\ 2.3)$$

These appear to form two clusters: a lower cluster near -1.8 and an upper cluster near 1.6.

Hyperparameters: $\alpha_0 = 1$ (uniform prior on π), $m = 0$, $s^2 = 10$ (diffuse prior on means), $\sigma^2 = 1$ (known noise variance).

Initialization: $\mu_1^{(0)} = 2.0$, $\mu_2^{(0)} = -2.0$, $\pi^{(0)} = 0.5$.

Iteration 1, Step 1: Update assignments.

Compute $\mathcal{N}(x_i; \mu_1^{(0)}, 1) = \mathcal{N}(x_i; 2.0, 1)$ and $\mathcal{N}(x_i; -2.0, 1)$ for each i. With $\pi^{(0)} = 0.5$, the responsibility simplifies to:

$$r_i^{(1)} = \frac{\mathcal{N}(x_i; 2.0, 1)}{\mathcal{N}(x_i; 2.0, 1) + \mathcal{N}(x_i; -2.0, 1)} = \frac{e^{-(x_i-2)^2/2}}{e^{-(x_i-2)^2/2} + e^{-(x_i+2)^2/2}} = \frac{1}{1 + e^{-4x_i}}$$

(using the identity $(x_i-2)^2 - (x_i+2)^2 = -8x_i$). This is the sigmoid function $\sigma(4x_i)$.

i	x_i	$r_i^{(1)}$	$z_i^{(1)}$ (draw)	Notes
1	-2.1	$\sigma(-8.4) \approx 0.000$	0	clearly comp. 2
2	-1.8	$\sigma(-7.2) \approx 0.001$	0	clearly comp. 2
3	-1.5	$\sigma(-6.0) \approx 0.002$	0	clearly comp. 2
4	-0.3	$\sigma(-1.2) \approx 0.232$	0	ambiguous
5	0.8	$\sigma(\ 3.2) \approx 0.961$	1	likely comp. 1
6	1.2	$\sigma(\ 4.8) \approx 0.992$	1	clearly comp. 1
7	1.9	$\sigma(\ 7.6) \approx 1.000$	1	clearly comp. 1
8	2.3	$\sigma(\ 9.2) \approx 1.000$	1	clearly comp. 1

Suppose the draws give $z^{(1)} = (0, 0, 0, 0, 1, 1, 1, 1)$. So $n_1^{(1)} = 4$, $n_0^{(1)} = 4$.

Iteration 1, Step 2: Update π.

$$\pi^{(1)} \sim \text{Beta}(1 + 4,\ 1 + 4) = \text{Beta}(5, 5)$$

Mean: $5/10 = 0.5$. A draw from Beta(5,5) might give $\pi^{(1)} \approx 0.48$.

Iteration 1, Step 3: Update μ_1 and μ_2.

Component 1 ($z_i = 1$): observations $\{0.8, 1.2, 1.9, 2.3\}$, $n_1 = 4$, $\bar{x}_1 = 6.2/4 = 1.55$.

$$\frac{1}{\tau_1^2} = \frac{1}{10} + \frac{4}{1} = 4.1, \quad \tau_1^2 = 0.244$$

$$\mu_1^* = 0.244\left(\frac{0}{10} + \frac{4 \times 1.55}{1}\right) = 0.244 \times 6.2 = 1.513$$

Draw: $\mu_1^{(1)} \sim \mathcal{N}(1.513, 0.244)$. A draw might give $\mu_1^{(1)} \approx 1.58$.

Component 2 ($z_i = 0$): observations $\{-2.1, -1.8, -1.5, -0.3\}$, $n_0 = 4$, $\bar{x}_0 = -5.7/4 = -1.425$.

$$\frac{1}{\tau_0^2} = \frac{1}{10} + \frac{4}{1} = 4.1, \quad \tau_0^2 = 0.244$$

$$\mu_2^* = 0.244\left(\frac{0}{10} + \frac{4 \times (-1.425)}{1}\right) = 0.244 \times (-5.7) = -1.391$$

Draw: $\mu_2^{(1)} \sim \mathcal{N}(-1.391, 0.244)$. A draw might give $\mu_2^{(1)} \approx -1.44$.

After one iteration we already have $\mu_1^{(1)} \approx 1.58$, $\mu_2^{(1)} \approx -1.44$, close to the true cluster centers. The chain has found the structure quickly.

After many iterations. The following table shows approximate posterior summaries after $T = 2000$ iterations with $T_{\text{burn}} = 500$:

Parameter	Prior mean	True value	Post. mean	Post. 95% CI
π	0.50	0.50	0.51	$[0.28, 0.74]$
μ_1	0.00	1.55	1.57	$[1.05, 2.10]$
μ_2	0.00	-1.43	-1.44	$[-1.93, -0.94]$

The posterior credible intervals are wide because $n = 8$ is small. With more data the intervals would narrow, and the posteriors would concentrate around the true values.

The Label Switching Problem

The two-component mixture has a fundamental symmetry: if we swap the labels ($1 \leftrightarrow 0$) and simultaneously swap ($\mu_1 \leftrightarrow \mu_2$) and replace π with $1-\pi$, the likelihood and prior are unchanged. The posterior has two identical modes: one with ($\mu_1 \approx 1.6, \mu_2 \approx -1.4$) and one with ($\mu_1 \approx -1.4, \mu_2 \approx 1.6$).

A well-mixing Gibbs sampler will eventually visit both modes, jumping between them. When it does, the marginal posterior of μ_1 is a bimodal distribution centered at both $+1.6$ and -1.4, with posterior mean near zero — completely useless as a point estimate.

Label switching is not a bug. The sampler is working correctly: the posterior genuinely has two symmetric modes. The fix is to post-process the samples by imposing an identifiability constraint. The simplest choice for our model:

$$\mu_1^{(t)} > \mu_2^{(t)} \quad \text{for all } t$$

After collecting samples, discard any where $\mu_1^{(t)} \leq \mu_2^{(t)}$ (or relabel: if $\mu_1^{(t)} \leq \mu_2^{(t)}$, swap $\mu_1^{(t)} \leftrightarrow \mu_2^{(t)}$, swap $z_i^{(t)}$ accordingly, and replace $\pi^{(t)}$ with $1-\pi^{(t)}$). This restores identifiability without biasing the posterior.

Implementation in Pseudocode

The following pseudocode implements the complete sampler. All standard probability distributions are assumed available (normal PDF, Bernoulli sampler, Beta sampler, Gaussian sampler).

```
Input:  data x[1..n], hyperparams alpha0, m, s2, sigma2
        T = total iterations, T_burn = burn-in length
Output: posterior samples {pi, mu1, mu2, z}[T_burn+1..T]

# Initialize
mu1 = mean(x) + std(x)
mu2 = mean(x) - std(x)
pi  = 0.5

# Main loop
for t = 1 to T:

    # Step 1: Update assignments z[i]
```

```
for i = 1 to n:
    l1 = pi    * NormalPDF(x[i], mu1, sigma2)
    l0 = (1-pi)* NormalPDF(x[i], mu2, sigma2)
    r  = l1 / (l1 + l0)
    z[i] = Bernoulli(r)

# Step 2: Update mixing weight pi
n1 = sum(z)
n0 = n - n1
pi = BetaSample(alpha0 + n1, alpha0 + n0)

# Step 3: Update component mean mu1
if n1 > 0:
    xbar1    = mean(x[i] for z[i]=1)
    prec1    = 1/s2 + n1/sigma2
    mu1_star = (m/s2 + n1*xbar1/sigma2) / prec1
    mu1      = NormalSample(mu1_star, 1/prec1)
else:
    mu1 = NormalSample(m, s2)  # draw from prior

# Step 4: Update component mean mu2
if n0 > 0:
    xbar0    = mean(x[i] for z[i]=0)
    prec0    = 1/s2 + n0/sigma2
    mu2_star = (m/s2 + n0*xbar0/sigma2) / prec0
    mu2      = NormalSample(mu2_star, 1/prec0)
else:
    mu2 = NormalSample(m, s2)  # draw from prior

# Enforce identifiability
if mu1 < mu2:
    swap(mu1, mu2)
    flip all z[i]: z[i] = 1 - z[i]
    pi = 1 - pi

# Store samples after burn-in
if t > T_burn:
    store(pi, mu1, mu2, z)
```

The cost per iteration is $O(n)$: one pass over the data for the assignment step, plus $O(1)$ work for the parameter updates. For $n = 1000$ and $T = 5000$ iterations, the total cost is five million simple operations — fast enough to run in seconds on modern hardware.

8.3 MCMC: Origins and Principles

Gibbs sampling is one member of a large family of algorithms called **Markov chain Monte Carlo (MCMC)**. To close this chapter, we place MCMC in its historical and conceptual context.

Origins in Statistical Physics

MCMC was invented not by statisticians but by physicists, and not for Bayesian inference but for computing thermodynamic quantities.

In 1953, Nicholas Metropolis, Arianna Rosenbluth, Marshall Rosenbluth, Augusta Teller, and Edward Teller published a paper titled "Equation of state calculations by fast computing machines" in the *Journal of Chemical Physics*. The problem was to compute thermodynamic properties — pressure, energy, specific heat — of a system of interacting particles. These properties are averages over the **Boltzmann distribution**:

$$p(\text{state}) \propto e^{-E(\text{state})/kT}$$

where E is the energy of the state, k is Boltzmann's constant, and T is temperature. This distribution assigns high probability to low-energy states and low probability to high-energy states, but its normalizing constant (the **partition function** $Z = \sum_{\text{states}} e^{-E/kT}$) is intractable for any system of realistic size.

The Metropolis algorithm circumvents this. Propose a small random change to the current state. If the change lowers the energy (increases probability), accept it. If it raises the energy by ΔE, accept it with probability $e^{-\Delta E/kT}$. This acceptance probability depends only on the ratio $p(\text{new})/p(\text{current}) = e^{-\Delta E/kT}$, not on the partition function. The resulting chain has the Boltzmann distribution as its stationary distribution.

The mathematical justification was provided by the **detailed balance** condition: a Markov chain has stationary distribution p if, for every pair of states x and y:

$$p(x)\,T(x \to y) = p(y)\,T(y \to x)$$

where $T(x \to y)$ is the transition probability. Detailed balance says the chain is in detailed equilibrium: the flow of probability from x to y equals the flow from y to x. Gibbs sampling satisfies detailed balance with respect to the target posterior, as does the Metropolis algorithm.

In 1970, W.K. Hastings generalized the Metropolis algorithm to arbitrary proposal distributions, giving the **Metropolis-Hastings** algorithm: the most general and widely used MCMC method. The acceptance probability for a proposed move from x to x' under proposal $q(x' \mid x)$ is:

$$\alpha(x, x') = \min\left(1,\ \frac{p(x')\,q(x \mid x')}{p(x)\,q(x' \mid x)}\right)$$

This ratio depends only on the unnormalized densities: $p(x')$ and $p(x)$ can be replaced by any positive functions proportional to the target, since the normalizing constant cancels in the ratio. This is the fundamental observation that makes MCMC applicable to Bayesian posteriors: we can evaluate $p(\theta \mid \mathcal{D}) \propto p(\mathcal{D} \mid \theta)p(\theta)$ without knowing the normalizing constant $p(\mathcal{D})$.

MCMC Comes to Bayesian Statistics

For three decades after 1953, MCMC remained confined to physics and chemistry. Its potential for Bayesian computation was recognized, but Bayesian methods were themselves rarely used — the computational obstacles were considered insurmountable.

Two developments changed this. First, the rapid increase in computing power through the 1980s made MCMC feasible for statistical problems of realistic size. Second, and more importantly, a 1990 paper by Gelfand and Smith in the *Journal of the American Statistical Association* pointed out that Gibbs sampling — rediscovered independently from the physics literature by the image analysis community (Geman and Geman, 1984, who coined the name) — provided a practical general algorithm for Bayesian computation in hierarchical models.

The 1990s saw an explosion of Bayesian MCMC methods. The WinBUGS software (Bayesian inference Using Gibbs Sampling), released in 1997, made MCMC accessible to applied statisticians and epidemiologists without requiring them to implement the algorithms from scratch. Stan, released in 2012, extended this to Hamiltonian Monte Carlo and became the dominant platform for modern Bayesian computation.

The MCMC Zoo

Modern MCMC encompasses a rich collection of algorithms, each suited to different posterior geometries.

Gibbs sampling (this chapter): update each coordinate from its full conditional, one at a time. Exact and efficient when conditionals are conjugate. Slow when posteriors are highly correlated.

Metropolis-Hastings: propose a move from a proposal distribution, accept or reject using the Metropolis-Hastings ratio. General-purpose but sensitive to the choice of proposal. Tuning the step size to achieve $\sim$23% acceptance is the classical recommendation (Roberts and Rosenthal, 2001).

Hamiltonian Monte Carlo (HMC): introduce auxiliary momentum variables and simulate Hamiltonian dynamics to propose large moves that respect the posterior geometry. Avoids random-walk behavior; mixes much faster than random-walk Metropolis in high dimensions. Requires the gradient of the log posterior (available for most statistical models). HMC is the engine of Stan.

No-U-Turn Sampler (NUTS): an adaptive HMC variant (Hoffman and Gelman, 2014) that automatically tunes the step size and trajectory length. The default sampler in Stan and PyMC.

Sequential Monte Carlo (SMC): maintains a population of particles that are reweighted and resampled as data arrives sequentially. Natural for time series and online learning.

Reversible jump MCMC: allows the dimension of the parameter space to change during the chain, enabling inference over models of different complexity (e.g., choosing the number of mixture components).

The unifying principle. Every MCMC algorithm constructs a Markov chain on the parameter space with the target posterior as its stationary distribution. The three ingredients are:

1. A **proposal mechanism** that suggests new parameter values from the current ones.
2. An **acceptance criterion** that ensures detailed balance (or a weaker condition called balance).
3. A **convergence guarantee** that the chain eventually reaches stationarity regardless of its starting point, under mild conditions (irreducibility and aperiodicity).

The normalizing constant of the posterior never needs to be computed: it cancels in every acceptance ratio. This is the fundamental reason MCMC makes Bayesian inference computationally tractable.

Diagnosing Convergence

A practical challenge with MCMC is determining when the chain has converged. No finite chain is exactly distributed as the posterior; we can only ask whether it is *approximately* so.

Trace plots. Plot $\theta^{(t)}$ against t for each parameter. A converged chain looks like white noise around a fixed level. A non-converged chain drifts, gets stuck, or has not yet reached the high-probability region.

The $\hat{R}$ statistic (Gelman-Rubin diagnostic): run multiple chains from different starting points. Compute the ratio of between-chain variance to within-chain variance. If $\hat{R} \approx 1$, the chains have mixed and are sampling from the same distribution. If $\hat{R} \gg 1$, the chains have not yet converged.

Effective sample size (ESS). Because consecutive MCMC samples are correlated, T samples from a chain contain less information than T independent samples. The ESS is:

$$\text{ESS} = \frac{T}{1 + 2\sum_{k=1}^{\infty} \rho_k}$$

where ρ_k is the lag-k autocorrelation. For a Gibbs sampler with high posterior correlation, ESS can be much smaller than T: the chain makes many correlated steps before

generating a truly independent sample.

Exercise 8.2. Write out the three conditional distributions for the two-component Gaussian mixture model when the variance σ^2 is also unknown, with prior $\sigma^2 \sim$ Inverse-Gamma(a_0, b_0). What is the conditional distribution $p(\sigma^2 \mid \mu_1, \mu_2, z, x)$? (Hint: the Inverse-Gamma distribution is conjugate to the Gaussian variance; recall that if $\sigma^2 \sim \mathrm{IG}(a, b)$ and $x_i \sim \mathcal{N}(\mu, \sigma^2)$, then $\sigma^2 \mid x \sim \mathrm{IG}(a + n/2,\ b + \sum (x_i - \mu)^2/2)$.)

Exercise 8.3. In the island picture, explain in your own words why the Gibbs shuffling procedure leaves the target distribution $p(x, y)$ invariant. Then show formally that the Gibbs transition kernel for a bivariate distribution,

$$T((x, y) \to (x', y)) = p(x' \mid y), \quad T((x, y) \to (x, y')) = p(y' \mid x)$$

satisfies the detailed balance condition $p(x, y)T((x, y) \to (x', y)) = p(x', y)T((x', y) \to (x, y))$.

Exercise 8.4. Implement the two-component Gaussian mixture Gibbs sampler in a programming language of your choice.

1. Generate $n = 200$ observations from a mixture of $\mathcal{N}(-2, 1)$ and $\mathcal{N}(2, 1)$ with equal weights ($\pi = 0.5$).

2. Run the Gibbs sampler for $T = 3000$ iterations with $T_{\text{burn}} = 1000$. Use hyperparameters $\alpha_0 = 1$, $m = 0$, $s^2 = 100$, $\sigma^2 = 1$.

3. Plot trace plots for π, μ_1, μ_2. Does the chain appear to have converged?

4. Compute posterior means and 95% credible intervals for all three parameters. Compare to the true values.

5. Plot a histogram of the data and overlay the fitted mixture density using the posterior mean parameters.

Chapter 9

Langevin Dynamics

Gibbs sampling works by slicing: it updates one coordinate at a time from its one-dimensional conditional distribution. This is elegant when the conditionals are conjugate, as in the Gaussian mixture model. But most posteriors in modern statistics — logistic regression, neural networks, any model with a non-Gaussian likelihood — have no conjugate structure. The conditionals belong to no standard family and cannot be sampled directly.

Langevin dynamics takes a different approach. Instead of slicing, it uses the *gradient* of the log posterior to steer a continuous random walk toward high-probability regions. It is a Markov chain on a continuous space, derived from a stochastic differential equation in continuous time and discretized for computation. And it requires nothing beyond the gradient of the log posterior — a quantity available for virtually any differentiable model.

9.1 Langevin Dynamics as Continuous-Space MCMC

The Setup

Let $p(\theta)$ be a probability density on $\mathbb{R}^d$ from which we wish to sample. In our application p will be the posterior $p(\theta \mid \mathcal{D})$, but for now we treat it as a general target density. We assume p is differentiable and that $\nabla \log p(\theta)$ is computable.

The **continuous-time Langevin stochastic differential equation** (SDE) is:

$$d\theta_t = \frac{1}{2}\nabla \log p(\theta_t)\, dt + dB_t$$

where B_t is a standard d-dimensional Brownian motion. This SDE has a remarkable property: its stationary distribution is exactly $p(\theta)$. The process, if run long enough, visits regions of $\mathbb{R}^d$ with frequency proportional to $p(\theta)$.

We cannot simulate the SDE exactly in continuous time. Instead, we discretize it with a small step size Δt. Over the interval $[t, t + \Delta t]$, the Brownian increment $B_{t+\Delta t} - B_t \sim \mathcal{N}(0, \Delta t\, I_d)$. Approximating the drift term as constant over the interval:

Discretized Langevin Update:

$$\theta_{t+\Delta t} = \theta_t + \frac{\Delta t}{2}\nabla \log p(\theta_t) + \sqrt{\Delta t}\,\xi_t, \qquad \xi_t \sim \mathcal{N}(0, I_d)$$

The update has two components:

- **Drift:** $\frac{\Delta t}{2}\nabla \log p(\theta_t)$. This is a small step of gradient ascent on the log density, pointing toward regions of higher probability.
- **Diffusion:** $\sqrt{\Delta t}\,\xi_t$. This is Gaussian noise with standard deviation $\sqrt{\Delta t}$ in each coordinate. It injects randomness, allowing the chain to explore.

Why does the noise scale as $\sqrt{\Delta t}$? This is not a convention. Brownian motion has the property that its displacement in time Δt has standard deviation $\sqrt{\Delta t}$, not Δt. This follows from the central limit theorem: a Brownian increment is the sum of many small independent steps, each of size $\sqrt{dt/n}$ for $n \to \infty$, and the sum of n such steps has standard deviation $\sqrt{n} \cdot \sqrt{dt/n} = \sqrt{dt}$. If the noise were $\Delta t\,\xi_t$ instead, it would be negligibly small relative to the drift as $\Delta t \to 0$, and the dynamics would collapse to deterministic gradient ascent, converging to a single mode rather than sampling the full distribution.

The One Million Particles Picture

Place one million particles in $\mathbb{R}^d$, distributed according to some initial density $q_0(\theta)$. At each step, every particle independently applies the Langevin update. The density of the population evolves over time. The question is: why does this population eventually distribute according to $p(\theta)$?

The answer comes from understanding what each component of the update does to the population density.

Component 1: Diffusion (adding noise). Each particle receives an independent Gaussian kick $\sqrt{\Delta t}\,\xi$. This spreads the population outward: any concentrated cluster disperses, and the population density becomes more uniform. Formally, adding independent $\mathcal{N}(0, \Delta t\, I_d)$ noise to every particle convolves the population density with a Gaussian kernel:

$$q \;\longmapsto\; q * \mathcal{N}(0, \Delta t\, I_d)$$

This always makes the density more diffuse: tighter concentrations spread out, sharp peaks smooth over, and the density moves toward uniformity.

Component 2: Drift (gradient ascent on log p). Each particle moves by $\frac{\Delta t}{2}\nabla \log p(\theta)$. Particles in low-density regions (where $\log p$ is increasing) move toward higher-density regions. Particles already in high-density regions (near a mode of p) experience small drift. This concentrates the population toward the high-probability regions of p: it makes the density more focused, the opposite of diffusion.

The balance principle. At stationarity, the spreading effect of diffusion and the concentrating effect of drift exactly cancel. The population density does not change from one step to the next. This cancellation defines the stationary distribution: it is the unique density p for which the two forces are in equilibrium.

The diffusion force always pushes toward uniformity. The drift force pushes toward the shape of $\log p$. Their equilibrium is p itself.

This is the physical picture behind Langevin dynamics. In statistical physics, the same equation describes a Brownian particle in a potential well $U(\theta) = -\log p(\theta)$: the particle is pushed toward the minimum of the potential (maximum of p) by the drift force and kicked by thermal noise. The temperature of the system determines the balance: high temperature means strong noise and near-uniform distribution; low temperature means weak noise and concentration near the mode. In our setting, the "temperature" is Δt (or 1 in the continuous-time SDE) and is fixed so that the equilibrium is exactly p.

9.2 Formal Justification via Test Functions

We now make the balance argument precise using **test functions**. This approach avoids working directly with density functions (which require solving partial differential equations) and instead works with expectations, which require only Taylor expansions.

The Test Function Approach

Let $h : \mathbb{R}^d \to \mathbb{R}$ be any smooth bounded function. We say the Langevin dynamics preserves a distribution p if, for every such h:

$$\mathbb{E}_p[h(\theta)] = \mathbb{E}_p[h(\theta')]$$

where $\theta \sim p$ and $\theta' = \theta + \frac{\Delta t}{2}\nabla \log p(\theta) + \sqrt{\Delta t}\,\xi$ is the updated position. In other words, if we start the particles at the stationary distribution p and apply one Langevin step, the expected value of any test function is unchanged.

We will show this holds to order Δt, confirming that p is the stationary distribution of the discretized dynamics in the limit $\Delta t \to 0$.

Notation

Write the Langevin update as:

$$\theta' = \theta + \underbrace{\frac{\Delta t}{2}\nabla \log p(\theta)}_{=:\,\delta(\theta)} + \underbrace{\sqrt{\Delta t}\,\xi}_{=:\,\eta}$$

where $\delta(\theta) = \frac{\Delta t}{2}\nabla \log p(\theta)$ is the deterministic drift and $\eta = \sqrt{\Delta t}\,\xi$ with $\xi \sim \mathcal{N}(0, I_d)$. We will compute $\mathbb{E}[h(\theta')]$ by expanding $h(\theta') = h(\theta+\delta+\eta)$ in a Taylor series around θ.

Taylor Expansion of $h(\theta')$

Expand $h(\theta + \delta + \eta)$ to second order:

$$\begin{aligned} h(\theta') &= h(\theta + \delta + \eta) \\ &= h(\theta) + (\delta + \eta)^\top \nabla h(\theta) \\ &\quad + \frac{1}{2}(\delta + \eta)^\top \nabla^2 h(\theta)\,(\delta + \eta) + O(\Delta t^{3/2}) \end{aligned}$$

Expand the quadratic term:

$$(\delta + \eta)^\top \nabla^2 h\,(\delta + \eta) = \delta^\top \nabla^2 h\,\delta + 2\delta^\top \nabla^2 h\,\eta + \eta^\top \nabla^2 h\,\eta$$

Now take expectations over ξ (and hence $\eta = \sqrt{\Delta t}\,\xi$), holding θ fixed.
First-order terms:

- $\mathbb{E}[\delta^\top \nabla h(\theta)] = \delta^\top \nabla h(\theta) = \frac{\Delta t}{2}(\nabla \log p(\theta))^\top \nabla h(\theta)$. This is $O(\Delta t)$.
- $\mathbb{E}[\eta^\top \nabla h(\theta)] = \mathbb{E}[\sqrt{\Delta t}\,\xi^\top]\nabla h(\theta) = 0$ since $\mathbb{E}[\xi] = 0$.

Second-order terms:

- $\mathbb{E}[\delta^\top \nabla^2 h\,\delta] = O(\Delta t^2)$: negligible.
- $\mathbb{E}[2\delta^\top \nabla^2 h\,\eta] = 0$ since $\mathbb{E}[\eta] = 0$.
- $\mathbb{E}[\eta^\top \nabla^2 h(\theta)\,\eta]$: since $\eta = \sqrt{\Delta t}\,\xi$ with $\xi \sim \mathcal{N}(0, I_d)$:

$$\mathbb{E}[\eta^\top \nabla^2 h\,\eta] = \Delta t\,\mathbb{E}[\xi^\top \nabla^2 h(\theta)\,\xi] = \Delta t\,\mathrm{tr}(\nabla^2 h(\theta)) = \Delta t\,\Delta h(\theta)$$

  where $\Delta h = \sum_{j=1}^d \partial^2 h/\partial \theta_j^2$ is the Laplacian of h. (We used the identity $\mathbb{E}[\xi^\top A \xi] = \mathrm{tr}(A)$ for $\xi \sim \mathcal{N}(0, I)$.)

Combining, and keeping only terms of order Δt:

$$\mathbb{E}_\xi[h(\theta')] = h(\theta) + \frac{\Delta t}{2}(\nabla \log p(\theta))^\top \nabla h(\theta) + \frac{\Delta t}{2}\Delta h(\theta) + O(\Delta t^{3/2})$$

Averaging Over $\theta \sim p$

Now take the expectation over $\theta \sim p$:

$$\mathbb{E}_p[h(\theta')] - \mathbb{E}_p[h(\theta)] = \frac{\Delta t}{2}\left[\mathbb{E}_p\big[(\nabla \log p(\theta))^\top \nabla h(\theta)\big] + \mathbb{E}_p[\Delta h(\theta)]\right] + O(\Delta t^{3/2})$$

We need to show the bracket is zero. Write $\mathbb{E}_p$ as an integral against $p(\theta)$:

$$\begin{aligned}&\mathbb{E}_p[(\nabla \log p)^\top \nabla h] + \mathbb{E}_p[\Delta h] \\ &= \int (\nabla \log p(\theta))^\top \nabla h(\theta)\, p(\theta)\, d\theta + \int \Delta h(\theta)\, p(\theta)\, d\theta\end{aligned}$$

Simplify the first term. Use $\nabla \log p = (\nabla p)/p$:

$$\int (\nabla \log p)^\top \nabla h \cdot p\, d\theta = \int (\nabla p)^\top \nabla h\, d\theta$$

Integrate by parts (in each coordinate j, integrate $\frac{\partial p}{\partial \theta_j}\frac{\partial h}{\partial \theta_j}$ by parts; boundary terms vanish since $p \to 0$ at infinity):

$$\int (\nabla p)^\top \nabla h\, d\theta = -\int p(\theta)\, \Delta h(\theta)\, d\theta = -\mathbb{E}_p[\Delta h]$$

Combining:

$$\mathbb{E}_p[(\nabla \log p)^\top \nabla h] + \mathbb{E}_p[\Delta h] = -\mathbb{E}_p[\Delta h] + \mathbb{E}_p[\Delta h] = 0$$

Stationarity result. For any smooth test function h:

$$\mathbb{E}_p[h(\theta')] = \mathbb{E}_p[h(\theta)] + O(\Delta t^{3/2})$$

The drift term (gradient ascent on $\log p$, contributing $-\mathbb{E}_p[\Delta h]$) and the diffusion term (Gaussian noise, contributing $+\mathbb{E}_p[\Delta h]$) exactly cancel. The distribution p is unchanged to leading order in Δt.

Why the two orders of Taylor expansion. The drift term $\delta = O(\Delta t)$ requires only a first-order expansion in δ: its contribution to $\mathbb{E}[h(\theta')]$ is $O(\Delta t)$. The diffusion term $\eta = O(\sqrt{\Delta t})$ requires a second-order expansion: its linear contribution $\mathbb{E}[\eta^\top \nabla h]$ vanishes by symmetry ($\mathbb{E}[\eta] = 0$), and its quadratic contribution $\frac{1}{2}\mathbb{E}[\eta^\top \nabla^2 h\, \eta] = \frac{\Delta t}{2}\Delta h$ is $O(\Delta t)$. Both contribute at the same order Δt, and their sum is zero. This exact matching of orders is not a coincidence: it is why the Langevin equation requires the specific scaling $\frac{\Delta t}{2}\nabla \log p$ for the drift (factor of $1/2$) paired with $\sqrt{\Delta t}$ for the noise.

9.3 Application: Bayesian Logistic Regression

We now apply Langevin dynamics to a concrete model with a non-conjugate posterior: Bayesian logistic regression. This is the canonical setting where no closed-form posterior exists and sampling methods are necessary.

The Model

We observe n training examples $(x_1, y_1), \ldots, (x_n, y_n)$ where $x_i \in \mathbb{R}^p$ is a feature vector and $y_i \in \{+1, -1\}$ is a binary label. The logistic regression model assigns the probability:

$$P(Y = +1 \mid x, w) = \sigma(w^\top x) = \frac{1}{1 + e^{-w^\top x}}$$

where $w \in \mathbb{R}^p$ is the weight vector and $\sigma(z) = 1/(1 + e^{-z})$ is the sigmoid function. Equivalently, the probability of the observed label y_i (which is ± 1) is:

$$P(Y = y_i \mid x_i, w) = \sigma(y_i \, w^\top x_i)$$

To see why: if $y_i = +1$, this gives $\sigma(w^\top x_i)$; if $y_i = -1$, this gives $\sigma(-w^\top x_i) = 1 - \sigma(w^\top x_i)$. Both cases are correct.

The Likelihood

Assuming the observations are conditionally independent given w:

$$p(\mathcal{D} \mid w) = \prod_{i=1}^{n} \sigma(y_i \, w^\top x_i)$$

Taking the log:

$$\log p(\mathcal{D} \mid w) = \sum_{i=1}^{n} \log \sigma(y_i \, w^\top x_i) = -\sum_{i=1}^{n} \log(1 + e^{-y_i \, w^\top x_i})$$

This is the **logistic log-likelihood**. It is a concave function of w, which means the likelihood surface has a single peak (no local maxima), but it is not Gaussian and has no conjugate prior.

The Prior

We place an isotropic Gaussian prior on w:

$$w \sim \mathcal{N}(0, \lambda^{-1} I_p)$$

where $\lambda > 0$ is the precision (inverse variance). The log prior is:

$$\log p(w) = -\frac{\lambda}{2} \|w\|^2 + C$$

This penalizes large weights, providing ℓ_2 regularization. As shown in Chapter 6, this is equivalent to ridge regression regularization.

The Log Posterior

By Bayes rule:

$$\log p(w \mid \mathcal{D}) = \log p(\mathcal{D} \mid w) + \log p(w) + C$$

$$\begin{aligned}\log p(w \mid \mathcal{D}) &= \sum_{i=1}^{n} \log \sigma(y_i\, w^\top x_i) - \frac{\lambda}{2}\|w\|^2 + C \\ &= -\sum_{i=1}^{n} \log(1 + e^{-y_i w^\top x_i}) - \frac{\lambda}{2}\|w\|^2 + C\end{aligned}$$

This is a concave function of w (sum of concave functions plus a quadratic penalty). The MAP estimate is the unique maximizer, obtainable by gradient ascent. But the full posterior is not Gaussian: the logistic log-likelihood curves differently from a quadratic, and the posterior has heavier tails than any Gaussian. To sample from the posterior we need MCMC.

The Score of the Posterior

The gradient of the log posterior with respect to w is the key quantity for Langevin dynamics. We compute it term by term.

Gradient of the log-likelihood. Differentiate $\log \sigma(y_i\, w^\top x_i)$ with respect to w:

$$\frac{\partial}{\partial w} \log \sigma(y_i\, w^\top x_i) = y_i\, x_i\, \sigma'(y_i\, w^\top x_i) / \sigma(y_i\, w^\top x_i)$$

Use the identity $\sigma'(z) = \sigma(z)(1 - \sigma(z))$:

$$= y_i\, x_i\, (1 - \sigma(y_i\, w^\top x_i))$$

Now use the identity $1 - \sigma(z) = \sigma(-z)$:

$$\frac{\partial}{\partial w} \log \sigma(y_i\, w^\top x_i) = y_i\, x_i\, \sigma(-y_i\, w^\top x_i)$$

Summing over all n observations:

$$\nabla_w \log p(\mathcal{D} \mid w) = \sum_{i=1}^{n} y_i\, x_i\, \sigma(-y_i\, w^\top x_i)$$

Gradient of the log prior.

$$\nabla_w \log p(w) = -\lambda w$$

Gradient of the log posterior.

$$\nabla_w \log p(w \mid \mathcal{D}) = \sum_{i=1}^{n} y_i \, x_i \, \sigma(-y_i \, w^\top x_i) - \lambda w$$

Interpreting the gradient. The weight on each training example i in the gradient is $\sigma(-y_i \, w^\top x_i)$: the probability that the current model gets example i *wrong*. When w correctly classifies x_i with high confidence (large $y_i \, w^\top x_i$), the weight is nearly zero and example i contributes little to the gradient. When w misclassifies x_i or is uncertain ($y_i \, w^\top x_i$ near zero), the weight is near 1 and example i pulls the gradient strongly. The gradient is automatically dominated by the hardest examples. The prior term $-\lambda w$ pulls the weights toward zero, providing regularization.

The Langevin Sampler for Bayesian Logistic Regression

Substituting into the Langevin update:

$$\begin{aligned} w^{(t+\Delta t)} &= w^{(t)} + \frac{\Delta t}{2} \nabla_w \log p(w^{(t)} \mid \mathcal{D}) + \sqrt{\Delta t} \, \xi^{(t)} \\ &= w^{(t)} + \frac{\Delta t}{2} \left[\sum_{i=1}^{n} y_i \, x_i \, \sigma(-y_i \, {w^{(t)}}^\top x_i) - \lambda w^{(t)} \right] + \sqrt{\Delta t} \, \xi^{(t)} \end{aligned}$$

where $\xi^{(t)} \sim \mathcal{N}(0, I_p)$.

The complete Langevin sampler for Bayesian logistic regression:

1. Initialize $w^{(0)}$ (e.g., at zero or at the MAP estimate).
2. At each step t:
   (a) For each training example i, compute the "error weight" $e_i = \sigma(-y_i \, {w^{(t)}}^\top x_i)$.
   (b) Compute the gradient: $g^{(t)} = \sum_{i=1}^{n} y_i \, x_i \, e_i - \lambda w^{(t)}$
   (c) Sample $\xi^{(t)} \sim \mathcal{N}(0, I_p)$.
   (d) Update: $w^{(t+\Delta t)} = w^{(t)} + \frac{\Delta t}{2} g^{(t)} + \sqrt{\Delta t} \, \xi^{(t)}$
3. After T_{burn} burn-in steps, collect samples $\{w^{(t)}\}$.

Cost per step: $O(np)$ for the gradient (one pass over the data), plus $O(p)$ for the noise. The same cost as gradient descent, but with deliberate noise added.

Step Size Selection

The step size Δt controls the accuracy-speed tradeoff:

- **Small Δt:** the discretization error is small, so the stationary distribution of the chain is close to the true posterior. But the chain moves slowly and requires many steps to explore the posterior.

- **Large Δt:** the chain moves quickly but the discretization error is large. The stationary distribution differs from the true posterior by $O(\Delta t)$.

A practical heuristic: set $\Delta t \propto p^{-1/3}$, where p is the dimension. This balances the step size against the dimension to maintain reasonable mixing.

Prediction from Posterior Samples

After collecting S samples $\{w^{(s)}\}_{s=1}^{S}$ from the posterior, predictions for a new input x_* use the **posterior predictive distribution**:

$$P(Y_* = +1 \mid x_*, \mathcal{D}) = \mathbb{E}_{w|\mathcal{D}}[\sigma(w^\top x_*)] \approx \frac{1}{S}\sum_{s=1}^{S} \sigma\Big({w^{(s)}}^\top x_*\Big)$$

This predictive probability is *not* the same as using the MAP estimate $\hat{w}_{\text{MAP}}$. Near the decision boundary (where $w^\top x_* \approx 0$ for most samples), the average $\frac{1}{S}\sum_s \sigma({w^{(s)}}^\top x_*)$ is close to $1/2$: genuine uncertainty. Far from the training data, the posterior over w is wide, and the predictions vary substantially across samples: again, genuine uncertainty. The MAP prediction $\sigma(\hat{w}_{\text{MAP}}^\top x_*)$ gives an overconfident single number without any measure of this uncertainty.

A Numerical Example

Consider a two-dimensional classification problem: $x_i \in \mathbb{R}^2$, $n = 20$ training points, prior precision $\lambda = 1$.

Data: 10 points from class $+1$ near $(2, 2)$ and 10 points from class -1 near $(-2, -2)$, with Gaussian scatter of standard deviation 1.

MAP estimate: gradient ascent on $\log p(w \mid \mathcal{D})$ converges to approximately $\hat{w}_{\text{MAP}} \approx (1.8, 1.8)^\top$ after 100 steps. The decision boundary $w^\top x = 0$ is the line $1.8x_1 + 1.8x_2 = 0$, i.e., $x_1 + x_2 = 0$.

Langevin sampler: run for $T = 5000$ steps with $\Delta t = 0.01$, burn-in $T_{\text{burn}} = 1000$.

After burn-in, the posterior samples $\{w^{(t)}\}$ concentrate around the MAP estimate with spread reflecting genuine posterior uncertainty. The following table shows posterior summaries:

Parameter	MAP	Post. mean	Post. std	95% CI
w_1	1.82	1.79	0.43	$[0.95, 2.63]$
w_2	1.81	1.78	0.42	$[0.96, 2.61]$

The posterior standard deviation of 0.43 is substantial: even with 20 training examples, the weights are not precisely determined. This uncertainty translates to prediction uncertainty: for a point at $(0.5, -0.5)$ (near the decision boundary), the Langevin predictive probability is ≈ 0.52 with a 95% predictive interval of $[0.31, 0.73]$, correctly reflecting that this point is genuinely ambiguous. The MAP prediction gives the single value $\sigma(0) = 0.5$ with no measure of uncertainty.

9.4 Stochastic Gradient Langevin Dynamics

For large datasets with $n \gg 1$, computing the full gradient $\sum_{i=1}^{n} y_i x_i \sigma(-y_i w^\top x_i)$ at every step costs $O(n)$ operations. This is the same cost as full-batch gradient descent and can be prohibitive when n is in the millions.

Stochastic gradient Langevin dynamics (SGLD), introduced by Welling and Teh (2011), replaces the full gradient with a minibatch gradient. At each step, sample a random minibatch $\mathcal{B}_t \subset \{1, \ldots, n\}$ of size $B \ll n$, and approximate:

$$\nabla_w \log p(\mathcal{D} \mid w) \approx \frac{n}{B} \sum_{i \in \mathcal{B}_t} y_i \, x_i \, \sigma(-y_i \, w^\top x_i)$$

The n/B scaling ensures the approximation is unbiased. The SGLD update is:

$$w^{(t+\Delta t)} = w^{(t)} + \frac{\Delta t}{2} \left[\frac{n}{B} \sum_{i \in \mathcal{B}_t} y_i \, x_i \, \sigma(-y_i \, {w^{(t)}}^\top x_i) - \lambda w^{(t)} \right] + \sqrt{\Delta t} \, \xi^{(t)}$$

The minibatch gradient introduces noise beyond the deliberate $\sqrt{\Delta t}\,\xi^{(t)}$ noise. When Δt is large, the deliberate noise is small relative to the minibatch noise, and SGLD behaves like stochastic gradient descent: it optimizes. When Δt is small, the deliberate noise dominates and SGLD behaves like a proper sampler. By annealing Δt from large to small, SGLD can transition from optimization to sampling, automatically locating the posterior mode and then exploring the posterior around it.

9.5 The Metropolis Correction

The discretized Langevin update introduces a bias of order Δt: the stationary distribution of the discrete chain is not exactly $p(\theta)$. For small Δt, this error is small and often acceptable. When exactness is required, the **Metropolis-Adjusted Langevin Algorithm (MALA)** adds a Metropolis-Hastings acceptance step after each Langevin proposal.

After proposing $\theta' = \theta + \frac{\Delta t}{2} \nabla \log p(\theta) + \sqrt{\Delta t}\,\xi$, accept θ' with probability:

$$\alpha(\theta, \theta') = \min\left(1, \; \frac{p(\theta') \, q(\theta \mid \theta')}{p(\theta) \, q(\theta' \mid \theta)}\right)$$

where $q(\theta' \mid \theta) = \mathcal{N}(\theta'; \theta + \frac{\Delta t}{2}\nabla \log p(\theta), \Delta t\, I)$ is the Langevin proposal density. If rejected, stay at θ. This Metropolis step exactly corrects the discretization error: the resulting chain has p as its exact stationary distribution, regardless of Δt.

The practical tradeoff: MALA is exact for any Δt but requires evaluating $p(\theta')$ (and its gradient for the reverse proposal), while unadjusted Langevin requires only the gradient. For posteriors where the bias of unadjusted Langevin is acceptable (small Δt, moderate d), the simpler version is preferred.

9.6 Comparison: Gibbs, Langevin, and Beyond

	Gibbs	**Langevin (ULA)**	**MALA**
Requirement	Closed-form conditionals	Gradient of $\log p$	Gradient of $\log p$
Step	Exact conditional draw	Gradient step + noise	Gradient step + noise + accept/reject
Exactness	Exact (in limit)	Biased ($O(\Delta t)$)	Exact
Conjugate models	Ideal	Redundant	Redundant
Non-conjugate	Requires MH steps	Natural	Natural
Large data	Requires full pass	Can use mini-batches (SGLD)	Harder with mini-batches

The three methods are complementary. Gibbs is the method of choice when all conditionals are conjugate: no gradient is required and each update is exact. Langevin and MALA are the methods of choice when the posterior is differentiable but non-conjugate: the gradient drives the chain toward the posterior efficiently, and MALA adds exactness at the cost of occasional rejection.

Exercise 9.1. Let $p(\theta) \propto e^{-\theta^4/4}$ on $\mathbb{R}$ (a non-Gaussian distribution with heavier tails than Gaussian).

1. Compute $\nabla \log p(\theta) = -\theta^3$ and write down the Langevin update.

2. Show using the test function argument that p is the stationary distribution of the Langevin dynamics: verify that $\mathbb{E}_p[(\nabla \log p)^\top \nabla h] + \mathbb{E}_p[\Delta h] = 0$ for $h(\theta) = \theta^2$ by direct integration.

3. Implement the Langevin sampler for $T = 10{,}000$ steps with $\Delta t = 0.01$ and plot a histogram of the samples. Overlay the true density $p(\theta) \propto e^{-\theta^4/4}$. Do the samples match?

Exercise 9.2. For Bayesian logistic regression with a single feature ($p = 1$), data $\{(x_i, y_i)\}_{i=1}^{10}$, and prior $w \sim \mathcal{N}(0, 1)$:

1. Write out the log posterior $\log p(w \mid \mathcal{D})$ explicitly as a function of w.
2. Compute the gradient $\nabla_w \log p(w \mid \mathcal{D})$ and verify the formula derived in the text.
3. Run the Langevin sampler for $T = 5000$ steps with $\Delta t = 0.05$. Plot the trace of $w^{(t)}$ and estimate the posterior mean and standard deviation.
4. Compare the posterior mean to the MAP estimate (obtained by running gradient ascent with no noise). Are they close? Why might they differ?

Exercise 9.3 (Stationarity of a Gaussian)**.** Let $p(\theta) = \mathcal{N}(\mu, \sigma^2)$ on $\mathbb{R}$. The score is $\nabla \log p(\theta) = -(\theta - \mu)/\sigma^2$.

1. Write down the Langevin update explicitly.
2. Show that if $\theta_t \sim \mathcal{N}(\mu, \sigma^2)$, then $\theta_{t+\Delta t}$ is also $\mathcal{N}(\mu, \sigma^2 + O(\Delta t^2))$ to leading order in Δt. (Hint: compute the mean and variance of $\theta_{t+\Delta t}$ using the linearity of the Gaussian score.)
3. What does this confirm about the stationary distribution of Langevin dynamics for a Gaussian target?

9.7 The Metropolis-Hastings Algorithm

The Metropolis correction of the previous section is a special case of a general principle. We now derive the **Metropolis-Hastings algorithm** in full generality, using the same one million people picture that has guided us throughout this book. The derivation requires no measure theory and no abstract algebra — only counting and the requirement that population flows balance.

The Setup: Proposed Migration and Visa Control

Let $p(x)$ be the target distribution we wish to sample from. Imagine one million people distributed according to $p(x)$: at stationarity, $p(x)$ million people are in state x, for every x.

We are free to choose any **proposal distribution** $q(y \mid x)$: the probability that a person in state x proposes to move to state y. The proposal is our design choice; it does not need to match p in any way.

Without any correction, allowing everyone to move according to $q(y \mid x)$ will generally destroy the distribution p: the flow of people leaving x for y will not balance the flow leaving y for x, and the population will drift away from p.

To preserve p, we introduce **visa control** at each proposed move. Think of it this way. Every person who proposes to move from x to y must apply for a visa at the *embassy of state y in state x*. The embassy may approve or deny the application. If denied, the person stays in x. The acceptance probability is chosen to ensure that the population remains distributed according to p.

Counting the Flows

At stationarity there are $p(x)$ million people in state x. Among them, a fraction $q(y \mid x)$ propose to move to y. The **proposed flow** from x to y is therefore:

$$F(x \to y) = p(x)\, q(y \mid x) \quad \text{(millions of people proposing to go from } x \text{ to } y\text{)}$$

Similarly, there are $p(y)$ million people in state y, of whom a fraction $q(x \mid y)$ propose to move to x. The proposed flow in the reverse direction is:

$$F(y \to x) = p(y)\, q(x \mid y)$$

For the population to remain at p, the actual flow from x to y must equal the actual flow from y to x: this is the **detailed balance** condition.

The Maximum Balanced Flow

What is the largest flow in both directions that can be allowed while maintaining balance? The actual flow from x to y cannot exceed the proposed flow $F(x \to y)$ — we cannot send more people than propose to go. And the two actual flows must be equal. Therefore the maximum balanced flow in both directions is:

$$a(x, y) = \min(F(x \to y),\ F(y \to x)) = \min(p(x)\, q(y \mid x),\ p(y)\, q(x \mid y))$$

This is the largest flow we can allow in each direction while keeping them equal.

The Acceptance Probability

The **acceptance probability** for a person proposing to move from x to y is the fraction of proposers who are granted a visa:

$$\alpha(x, y) = \frac{a(x, y)}{F(x \to y)} = \frac{\min(p(x)\, q(y \mid x),\ p(y)\, q(x \mid y))}{p(x)\, q(y \mid x)}$$

$$= \min\left(1,\ \frac{p(y)\, q(x \mid y)}{p(x)\, q(y \mid x)}\right)$$

The Metropolis-Hastings acceptance probability:

$$\alpha(x, y) = \min\left(1,\ \frac{p(y)\, q(x \mid y)}{p(x)\, q(y \mid x)}\right)$$

To propose a move from x to y: draw $y \sim q(y \mid x)$, then accept the move with probability $\alpha(x, y)$. If rejected, stay at x. The resulting Markov chain has p as its stationary distribution.

Why This Works: Detailed Balance by Construction

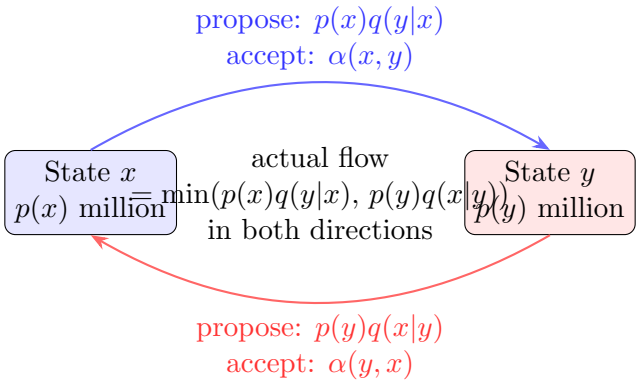
The actual flow from x to y after visa control is:

$$p(x)\, q(y \mid x)\, \alpha(x, y) = p(x)\, q(y \mid x) \cdot \min\left(1,\ \frac{p(y)\, q(x \mid y)}{p(x)\, q(y \mid x)}\right) = \min(p(x)\, q(y \mid x),\ p(y)\, q(x \mid y)) = a(x, y)$$

The actual flow from y to x is the same $a(x, y)$ by the same calculation with x and y swapped. Therefore:

$$p(x)\, q(y \mid x)\, \alpha(x, y) = p(y)\, q(x \mid y)\, \alpha(y, x)$$

This is **detailed balance**: the flow of people from x to y exactly equals the flow from y to x, at every pair of states. Detailed balance guarantees that p is the stationary distribution of the chain. The Metropolis-Hastings algorithm does not discover detailed balance as a theorem; it *constructs* it, by design, through the visa acceptance rule.



The Normalizing Constant Cancels

A crucial practical observation: the acceptance probability depends on the *ratio* $p(y)/p(x)$, not on $p(x)$ and $p(y)$ individually. If $p(x) = \tilde{p}(x)/Z$ for some unnormalized density $\tilde{p}$ and unknown constant Z:

$$\alpha(x, y) = \min\left(1,\ \frac{\tilde{p}(y)/Z \cdot q(x \mid y)}{\tilde{p}(x)/Z \cdot q(y \mid x)}\right) = \min\left(1,\ \frac{\tilde{p}(y)\, q(x \mid y)}{\tilde{p}(x)\, q(y \mid x)}\right)$$

The normalizing constant Z cancels in the ratio. This is the key property that makes Metropolis-Hastings applicable to Bayesian posteriors: we can evaluate $p(\theta \mid \mathcal{D}) \propto p(\mathcal{D} \mid \theta)\, p(\theta)$ without computing the intractable evidence $p(\mathcal{D})$, because it cancels in the acceptance ratio.

Special Cases

Symmetric proposal: the original Metropolis algorithm. If $q(y \mid x) = q(x \mid y)$ for all x, y (e.g., $q(y \mid x) = \mathcal{N}(y; x, \sigma^2 I)$, which is symmetric in x and y), the acceptance probability simplifies to:

$$\alpha(x, y) = \min\left(1,\ \frac{p(y)}{p(x)}\right)$$

Accept a move to higher probability unconditionally; accept a move to lower probability with probability equal to the probability ratio. This is the original Metropolis et al. (1953) rule.

Independent proposal. If $q(y \mid x) = q(y)$ does not depend on x at all, the acceptance probability becomes:

$$\alpha(x, y) = \min\left(1,\ \frac{p(y)\, q(x)}{p(x)\, q(y)}\right) = \min\left(1,\ \frac{p(y)/q(y)}{p(x)/q(x)}\right)$$

Accept if the proposed state y has a higher *importance weight* $p(y)/q(y)$ than the current state. This is the **independence sampler**; it works well when q is a good global approximation to p.

Langevin proposal: MALA. The Metropolis-Adjusted Langevin Algorithm uses the Langevin proposal:

$$q(y \mid x) = \mathcal{N}\left(y;\ x + \frac{\Delta t}{2} \nabla \log p(x),\ \Delta t\, I\right)$$

The acceptance probability corrects the discretization error of the Langevin update exactly: the chain has p as its stationary distribution for any step size Δt, not just in the limit $\Delta t \to 0$.

Gibbs sampling. Gibbs sampling is a special case of Metropolis-Hastings in which moves are always accepted: $\alpha(x, y) = 1$. This occurs when the proposal is the exact conditional distribution $q(y_j \mid x) = p(y_j \mid x_{-j})$: in that case $p(x)\, q(y \mid x) = p(x)\, p(y_j \mid x_{-j}) = p(x, y_j) = p(y)\, p(x_j \mid y_{-j}) = p(y)\, q(x \mid y)$, so the proposed flows are equal and every proposal is accepted. The embassy never issues a denial when the proposal is the exact conditional: the population is already in perfect balance.

The Metropolis-Hastings family.

Algorithm	**Proposal** $q(y\|x)$	**Acceptance** $\alpha(x, y)$
Metropolis (1953)	Symmetric: $q(y\|x) = q(x\|y)$	$\min(1,\, p(y)/p(x))$
Metropolis-Hastings (1970)	Any $q(y\|x)$	$\min(1,\, p(y)q(x\|y)/p(x)q(y\|x))$
Independence sampler	$q(y\|x) = q(y)$	$\min(1,\, [p(y)/q(y)]/[p(x)/q(x)])$
MALA	Langevin step from x	Full MH ratio with Langevin q
Gibbs	$q(y_j\|x) = p(y_j\|x_{-j})$	Always 1

Every algorithm in the table is the same embassy system with a different proposal distribution. The visa rule is always $\min(1, p(y)q(x|y)/p(x)q(y|x))$. The choice of proposal determines how efficiently the chain explores the target distribution; the acceptance rule ensures that exploration is always unbiased.

Choosing the Proposal

The proposal distribution $q(y \mid x)$ is the analyst's design choice, and it determines the practical performance of the algorithm.

Step size. For a random walk proposal $q(y \mid x) = \mathcal{N}(y; x, \sigma^2 I)$, the step size σ controls the tradeoff:

- Too small: proposals are nearly always accepted, but the chain moves slowly, making tiny steps. The chain needs many iterations to traverse the target distribution.
- Too large: proposals jump far away, but are frequently rejected because $p(y) \ll p(x)$. The chain spends most steps staying in place.

The optimal acceptance rate for a d-dimensional Gaussian target under a Gaussian random walk proposal is approximately 23.4%, achieved by setting $\sigma^2 \approx 2.38^2/d$ (Roberts, Gelman, and Gilks, 1997). This result guides tuning in practice.

Informed proposals. Any distribution can serve as a proposal. Proposals that incorporate information about p — such as the Langevin proposal, which uses the gradient of $\log p$ to bias moves toward higher probability — lead to higher acceptance rates and faster mixing. The visa analogy is apt: a traveler who applies for a visa to a country they are likely to be admitted to wastes less time on rejected applications.

Exercise 9.4. Let $p(x) \propto e^{-|x|}$ (the Laplace distribution on $\mathbb{R}$) and consider the symmetric proposal $q(y \mid x) = \mathcal{N}(y; x, \sigma^2)$.

1. Write down the Metropolis acceptance probability $\alpha(x, y) = \min(1, p(y)/p(x))$ explicitly in terms of $|y| - |x|$.

2. Show that a proposed move that decreases $|x|$ (moves toward the origin) is always accepted, while a move that increases $|x|$ is accepted with probability $e^{-(|y|-|x|)}$.

3. Explain in terms of the visa analogy why moves toward the high-density region are always granted a visa, while moves away from it face an exponentially decaying acceptance rate.

Exercise 9.5. Show that Gibbs sampling is a special case of Metropolis-Hastings with acceptance probability 1. Specifically, consider updating coordinate j: the current state is x, the proposal is $y = (x_{-j}, y_j)$ with $y_j \sim p(y_j \mid x_{-j})$, and $y_{-j} = x_{-j}$. Show that $p(x)\, q(y \mid x) = p(y)\, q(x \mid y)$ and conclude that $\alpha(x, y) = 1$.

Exercise 9.6. Consider an independence sampler with proposal $q(y) = \mathcal{N}(y; \mu, \tau^2)$ targeting $p(x) = \mathcal{N}(x; 0, 1)$.

1. Write down the acceptance probability $\alpha(x, y)$ in terms of the importance weights $w(x) = p(x)/q(x)$.

2. Show that if $\mu = 0$ and $\tau^2 = 1$ (proposal equals target), then $\alpha(x, y) = 1$ always. Why does this make sense in the visa analogy?

3. Show that if $\tau^2 \gg 1$ (proposal much wider than target), the chain mixes slowly because most of the time the proposed y has a much lower importance weight than the current x. Explain in terms of the embassy: what does a wide proposal mean for the visa approval rates?

9.8 The Genesis of the Metropolis Algorithm

The Metropolis-Hastings algorithm is named after a 1953 paper whose origin story is unlike that of almost any other foundational result in statistics. It was not written by statisticians. It was not motivated by inference. It was written by nuclear weapons physicists having fun with the most powerful computer in the world, in the aftermath of the most destructive weapons program in history.

Los Alamos and the First Computers

The Los Alamos National Laboratory was created in 1943 as the heart of the Manhattan Project: the secret American program to build the atomic bomb. The physicists assembled there — Fermi, Bethe, Feynman, Teller, Oppenheimer, and dozens of others — faced enormous computational demands. The implosion design for the plutonium bomb required solving equations describing shock waves, neutron transport, and hydrodynamics in three dimensions. Human computers — rooms full of people operating mechanical calculators in shifts — were not fast enough. The demand for computation was one of the driving forces behind the development of electronic computers.

Nicholas Metropolis was a Greek-American physicist who arrived at Los Alamos in 1943 as a young man of twenty-six. Unlike most of his colleagues, whose primary identity was as physicists or mathematicians, Metropolis became captivated by the machines themselves. After the war, he returned to Los Alamos with a singular mission: to build the most powerful computer in the world. The result was **MANIAC** — Mathematical Analyzer, Numerical Integrator, and Computer — completed in 1952. It occupied an entire room, weighed several tons, and could perform thousands of operations per second: breathtaking speed by the standards of its time. Metropolis ran the computing division at Los Alamos and controlled access to MANIAC. In the early 1950s, that made him one of the most important people in computational science.

After the Bomb: Teller and the Hydrogen Weapon

The atomic bomb had ended the war in 1945, but the physics establishment at Los Alamos did not disperse. Edward Teller, a Hungarian-American physicist who had been at Los Alamos during the Manhattan Project, had spent much of his time there pursuing a private obsession: a weapon far more powerful than the fission bomb, one based on nuclear fusion. He called it the Super. After the Soviet Union tested its own fission bomb in 1949, the American government committed to building it. Teller, finally vindicated, became the central figure in what would become the hydrogen bomb program. The first thermonuclear device, *Ivy Mike*, was detonated in November 1952 with a yield of ten megatons — five hundred times the bomb that had destroyed Hiroshima.

With the hydrogen bomb designed and tested, Teller and his colleagues found themselves at Los Alamos with extraordinary computational resources and, for the moment, somewhat less existential pressure. The MANIAC was up and running. The question became: what else could one do with it?

An Afternoon's Amusement

The answer came from **Marshall Rosenbluth**, a young physicist who had worked on the hydrogen bomb and was widely regarded as one of the most gifted theoretical physicists of his generation. (Hans Bethe later called him the "best physicist in America.") Rosenbluth was interested in statistical mechanics and the behavior of interacting particle systems. The central computational challenge was the one we have already described: computing thermodynamic averages over the Boltzmann distribution requires summing over an astronomically large number of configurations.

Rosenbluth proposed the following idea. Instead of enumerating configurations, simulate a random walk through configuration space. Propose a small random change to the current configuration. If the change lowers the energy, accept it. If it raises the energy, accept it with a probability that depends on the energy increase — high enough to allow thermal fluctuations, low enough to keep the walk in the neighborhood of the low-energy configurations that actually matter. The walk, after running long enough,

would sample configurations with frequency proportional to their Boltzmann weight. Averages could then be estimated from the trajectory.

This is the algorithm we derived in the previous section from the visa analogy. The insight that random acceptance of energy-raising moves — rather than always moving downhill — was necessary to sample the full distribution rather than merely optimize it was Rosenbluth's key contribution. A greedy algorithm that always lowered energy would converge to a local minimum, not a thermal equilibrium. The probabilistic acceptance rule was what distinguished sampling from optimization.

The Five-Author Paper and Two Couples

The paper that resulted, published in the *Journal of Chemical Physics* in 1953, had five authors: **Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller**. It was titled "Equation of State Calculations by Fast Computing Machines."

The authorship is remarkable for its social structure. Marshall Rosenbluth had recently married **Arianna Wright**, who was herself a physicist of considerable ability. She wrote the actual code that ran on MANIAC — in the machine language of the day, a painstaking and skilled task. **Edward Teller** and his wife **Augusta** (universally known as Mici) completed the second couple. The paper was thus, unusually, the collaborative product of two married pairs of physicists plus Metropolis, who contributed the computer.

Metropolis's name appears first despite his not having proposed the algorithm. This was not theft; it was pragmatics. Metropolis ran the computing division and controlled access to MANIAC. Without him, the paper could not have been written: there was no other machine on which the calculations could have been run, and access to MANIAC required his cooperation and institutional support. The convention of listing the laboratory director first was standard at Los Alamos. The result is that one of the most cited algorithms in the history of science bears the name of the man who provided the computer rather than the man who invented the method.

Marshall Rosenbluth was characteristically gracious about this for the rest of his life, though historians of science have increasingly credited him with the central idea. In a 2003 interview, reflecting on the fiftieth anniversary of the paper, he described the work as having been done almost as a recreation — a playful application of the new computing power to a problem that interested him personally. The algorithm was not the result of a funded program or a deliberate research agenda. It was, in his telling, what happened when brilliant people had access to a remarkable machine and some free time.

What They Were Computing

The application in the original paper was the equation of state of a two-dimensional system of hard disks: particles that repel each other when they overlap and ignore

each other otherwise. This is a classical model in statistical mechanics, related to the behavior of liquids and gases. The Boltzmann distribution over disk configurations at a given temperature determines the thermodynamic properties: pressure, density, energy, and so forth.

The paper reported results for 224 particles — a tiny system by modern standards, but a remarkable computation for 1952. The authors noted that the method was completely general: it could be applied to any system with a computable energy function, regardless of the number of particles or the form of the interactions. This generality was what made the paper so influential.

The Long Sleep and the Rediscovery

Despite its significance, the Metropolis algorithm spent nearly four decades as a tool known primarily to physicists and chemists. Statisticians were largely unaware of it. The Bayesian statistical community, for its part, had different problems: the computational obstacles to Bayesian inference were severe, but the dominant view was that they were insurmountable in principle, not merely in practice.

The connection between the Metropolis algorithm and Bayesian computation was made explicit in 1970 by W.K. Hastings, who generalized the acceptance rule to non-symmetric proposals — producing the Metropolis-Hastings algorithm as we now know it — and noted its applicability to statistical problems. But the revolution in Bayesian computation did not arrive until 1990, when Gelfand and Smith published their paper on Gibbs sampling and pointed out that MCMC methods could make Bayesian inference in hierarchical models computationally tractable. Within a decade, MCMC had transformed Bayesian statistics from a theoretical program with few practical applications into the dominant computational framework for statistical inference.

A remarkable provenance. The algorithm that underlies virtually all modern Bayesian computation was invented at a nuclear weapons laboratory, by physicists who had just built the hydrogen bomb, running on a computer that had been built to design nuclear weapons, written up as a five-author paper by two couples having what one of them later described as recreational fun. The visa analogy of the previous section — embassies, applications, balanced flows — is perhaps a more peaceful way to think about it. But the algorithm itself was born in one of the most consequential and morally complex moments in the history of science, which is not the worst reminder that the tools of inference are never entirely separable from the world that produces them.

The Most Important Algorithm of the Twentieth Century

In the year 2000, the editors of *Computing in Science and Engineering*, a joint publication of the American Institute of Physics and the IEEE Computer Society, compiled a list of the ten algorithms that had most shaped the practice of science and engineering

during the twentieth century. The list was assembled by Jack Dongarra and Francis Sullivan, two of the most respected figures in scientific computing, and the selections were debated and vetted by a panel of experts across mathematics, physics, computer science, and engineering.

The ten algorithms selected were: the simplex method for linear programming, the Krylov subspace iteration methods, the decompositional approach to matrix computations, the Fortran compiler, the QR algorithm for eigenvalue computation, the Quicksort algorithm, fast Fourier transform, the integer relation detection algorithm, the fast multipole method — and the Metropolis algorithm.

The citation for the Metropolis algorithm noted that it was "the most influential algorithm ever written in the physical sciences" and that it had, over the intervening half-century, become indispensable not only in statistical physics and chemistry but in fields its inventors could not have imagined: Bayesian statistics, machine learning, operations research, computational biology, and finance.

The recognition was belated by nearly fifty years, but it was unambiguous. The afternoon's amusement at Los Alamos — two couples, a physicist with a computer, and a problem about hard disks — had produced something that touched essentially every corner of computational science. Dongarra and Sullivan's commentary observed that what made the algorithm so durable was precisely its simplicity: the acceptance rule fits in a single line, requires no special structure of the target distribution beyond the ability to evaluate it pointwise, and scales to problems of arbitrary complexity. These are exactly the properties that made it the engine of the Bayesian computational revolution forty years after it was written.

Marshall Rosenbluth, who by 2000 was in his mid-seventies and had spent his career making foundational contributions to plasma physics, lived to see the recognition. Nicholas Metropolis had died in 1999, one year before the list was published, and so did not. Arianna Rosenbluth had left physics for other pursuits decades earlier and has reflected in interviews that she was largely unaware of the algorithm's subsequent fame until well into her later years. The five authors who sat down at Los Alamos to write what they thought of as a short technical note about hard disks could not have known they were writing one of the most consequential pages in the history of scientific computing.

Further Reading

The history of the Metropolis algorithm has been documented with increasing care as its importance has grown. Metropolis himself wrote a personal account in "The Beginning of the Monte Carlo Method" (*Los Alamos Science*, 1987), which is readable and charming. Arianna Rosenbluth's role in writing the code was underappreciated for decades and has been more fully recognized recently. The intellectual biography of Marshall Rosenbluth — who went on to make foundational contributions to plasma physics before his death in 2003 — remains to be written. The broader story of computation at Los Alamos, and its role in the development of modern scientific computing,

is told in George Dyson's *Turing's Cathedral* (2012).

Chapter 10

Variational Inference

The previous two chapters developed sampling-based approaches to Bayesian computation. Gibbs sampling and Langevin dynamics both produce approximate draws from the posterior; given enough time, they are asymptotically exact. Their cost is time: a complex posterior may require millions of steps before it is adequately explored.

Variational inference (VI) takes a fundamentally different approach. Instead of sampling from the posterior, it approximates the posterior with a tractable distribution from a chosen family, finding the best approximation by solving an optimization problem. The result is fast and scalable — often orders of magnitude faster than MCMC — but at the cost of accuracy: the approximation may be systematically biased in ways that sampling methods are not.

We begin with a concrete and complete example, derive the general theory, trace the method's origins in statistical physics, and then connect it to the modern landscape of generative models.

10.1 Motivating Example: Bayesian Logistic Regression

We return to the Bayesian logistic regression model from the previous chapter, now with the goal of approximating the posterior rather than sampling from it.

Setup

We have n training examples (x_i, y_i) with $x_i \in \mathbb{R}^p$ and $y_i \in \{+1, -1\}$. The model is:

$$P(Y = y_i \mid x_i, w) = \sigma(y_i\, w^\top x_i), \qquad w \sim \mathcal{N}(0, \lambda^{-1} I_p)$$

The log posterior is:

$$\log p(w \mid \mathcal{D}) = \sum_{i=1}^{n} \log \sigma(y_i\, w^\top x_i) - \frac{\lambda}{2}\|w\|^2 + C$$

This is a smooth concave function of w with a single mode, but it is not Gaussian. The posterior has heavier tails than any Gaussian because the logistic likelihood falls off more slowly than a quadratic. No conjugate posterior exists.

The Variational Family

We approximate the posterior with a Gaussian:

$$q(w; \mu, \Sigma) = \mathcal{N}(w; \mu, \Sigma)$$

For scalability we often restrict to a diagonal covariance: $\Sigma = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_p^2)$, so the variational parameters are $\phi = (\mu, \sigma_1, \ldots, \sigma_p)$, a total of $2p$ numbers. We want to find ϕ^* such that $q(w; \phi^*)$ is as close as possible to the true posterior $p(w \mid \mathcal{D})$.

The Objective: ELBO

The quality of the approximation is measured by the KL divergence from q to the posterior:

$$D_{\mathrm{KL}}(q_\phi \| p(\cdot \mid \mathcal{D})) = \mathbb{E}_{q_\phi}\left[\log \frac{q(w;\phi)}{p(w \mid \mathcal{D})}\right]$$

This is always non-negative and equals zero if and only if $q = p$ almost everywhere. We want to minimize it over ϕ.

The difficulty is that $p(w \mid \mathcal{D})$ involves the intractable normalizing constant $p(\mathcal{D})$. We expand:

$$\begin{aligned} D_{\mathrm{KL}}(q_\phi \| p(\cdot \mid \mathcal{D})) &= \mathbb{E}_{q_\phi}[\log q(w;\phi)] - \mathbb{E}_{q_\phi}[\log p(w \mid \mathcal{D})] \\ &= \mathbb{E}_{q_\phi}[\log q(w;\phi)] - \mathbb{E}_{q_\phi}[\log p(w, \mathcal{D})] + \log p(\mathcal{D}) \end{aligned}$$

Since $\log p(\mathcal{D})$ does not depend on ϕ, minimizing D_{KL} is equivalent to maximizing:

$$\mathcal{L}(\phi) = \mathbb{E}_{q_\phi}[\log p(w, \mathcal{D})] - \mathbb{E}_{q_\phi}[\log q(w;\phi)]$$

This is the **Evidence Lower BOund (ELBO)**. Writing $p(w, \mathcal{D}) = p(\mathcal{D} \mid w)\, p(w)$:

$$\begin{aligned} \mathcal{L}(\phi) &= \mathbb{E}_{q_\phi}[\log p(\mathcal{D} \mid w)] + \mathbb{E}_{q_\phi}[\log p(w)] - \mathbb{E}_{q_\phi}[\log q(w;\phi)] \\ &= \underbrace{\mathbb{E}_{q_\phi}[\log p(\mathcal{D} \mid w)]}_{\text{expected log-likelihood}} - \underbrace{D_{\mathrm{KL}}(q_\phi \| p(w))}_{\text{KL from prior}} \end{aligned}$$

The ELBO has a clean interpretation: it rewards q_ϕ for concentrating mass where the likelihood is high (fit to data) and penalizes it for deviating from the prior (complexity).

For the Gaussian prior $p(w) = \mathcal{N}(0, \lambda^{-1} I)$ and diagonal Gaussian $q_\phi = \mathcal{N}(\mu, \Sigma)$, the KL divergence is available in closed form:

$$D_{\mathrm{KL}}(q_\phi \| p(w)) = \frac{1}{2}\left[\lambda \|\mu\|^2 + \lambda \sum_j \sigma_j^2 - \sum_j \log(\lambda \sigma_j^2) - p\right]$$

The expected log-likelihood $\mathbb{E}_{q_\phi}[\log p(\mathcal{D} \mid w)]$ does not have a closed form for logistic regression. We estimate it by Monte Carlo.

The Reparameterization Trick

To maximize $\mathcal{L}(\phi)$ by gradient ascent, we need $\nabla_\phi \mathcal{L}(\phi)$. The KL term is analytic. For the expected log-likelihood, we need:

$$\nabla_\phi \, \mathbb{E}_{q_\phi}[\log p(\mathcal{D} \mid w)] = \nabla_\phi \int \log p(\mathcal{D} \mid w) \, q(w; \phi) \, dw$$

The difficulty: the distribution we are integrating over, $q(w; \phi)$, depends on ϕ. We cannot simply move the gradient inside the integral.

The **reparameterization trick** resolves this by expressing the sample $w \sim q(w; \phi)$ as a deterministic function of ϕ and a noise variable ϵ that is independent of ϕ. For the diagonal Gaussian:

$$w = g(\epsilon, \phi) = \mu + \Sigma^{1/2} \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I_p)$$

where $\Sigma^{1/2} = \mathrm{diag}(\sigma_1, \ldots, \sigma_p)$. Componentwise: $w_j = \mu_j + \sigma_j \epsilon_j$.

Now rewrite the expectation:

$$\mathbb{E}_{q_\phi}[\log p(\mathcal{D} \mid w)] = \mathbb{E}_\epsilon[\log p(\mathcal{D} \mid \mu + \Sigma^{1/2} \epsilon)]$$

The distribution of ϵ is $\mathcal{N}(0, I_p)$, which does not depend on ϕ. The gradient can now move inside:

$$\nabla_\phi \, \mathbb{E}_{q_\phi}[\log p(\mathcal{D} \mid w)] = \mathbb{E}_\epsilon[\nabla_\phi \log p(\mathcal{D} \mid \mu + \Sigma^{1/2} \epsilon)]$$

This expectation is estimated by drawing S samples $\epsilon^{(1)}, \ldots, \epsilon^{(S)} \sim \mathcal{N}(0, I_p)$:

$$\nabla_\phi \, \mathbb{E}_{q_\phi}[\log p(\mathcal{D} \mid w)] \approx \frac{1}{S} \sum_{s=1}^{S} \nabla_\phi \log p\big(\mathcal{D} \mid \mu + \Sigma^{1/2} \epsilon^{(s)}\big)$$

For the log-likelihood of logistic regression, the gradient with respect to μ and σ_j is:

$$\frac{\partial}{\partial \mu} \log p(\mathcal{D} \mid w^{(s)}) = \nabla_w \log p(\mathcal{D} \mid w)\big|_{w=w^{(s)}}$$

$$\frac{\partial}{\partial \sigma_j} \log p(\mathcal{D} \mid w^{(s)}) = \left.\frac{\partial \log p(\mathcal{D} \mid w)}{\partial w_j}\right|_{w=w^{(s)}} \cdot \epsilon_j^{(s)}$$

where the gradient of the log-likelihood with respect to w is $\sum_{i=1}^{n} y_i x_i \sigma(-y_i w^\top x_i)$, as derived in the previous chapter.

The reparameterization trick. By writing $w = \mu + \Sigma^{1/2}\epsilon$ with $\epsilon \sim \mathcal{N}(0, I_p)$:

1. The randomness in w is separated from the variational parameters $\phi = (\mu, \Sigma^{1/2})$.
2. Gradients of the ELBO with respect to ϕ can be computed by backpropagation through the deterministic map $g(\epsilon, \phi)$.
3. The gradient estimator has low variance because the randomness (ϵ) is independent of the parameters being optimized.

Without reparameterization, the gradient estimator has much higher variance (the REINFORCE estimator) and optimization is much slower. The reparameterization trick is the key computational innovation that makes variational inference practical for continuous latent variables.

The Full Algorithm

Variational Bayes for Logistic Regression:
Variational family: $q(w; \phi) = \mathcal{N}(w; \mu, \mathrm{diag}(\sigma_1^2, \ldots, \sigma_p^2))$.
ELBO: $\mathcal{L}(\phi) = \mathbb{E}_{q_\phi}[\log p(\mathcal{D} \mid w)] - D_{\mathrm{KL}}(q_\phi \| p(w))$
At each gradient step:

1. Draw $S = 5$–10 noise samples $\epsilon^{(s)} \sim \mathcal{N}(0, I_p)$.
2. Form $w^{(s)} = \mu + \Sigma^{1/2}\epsilon^{(s)}$.
3. Compute stochastic gradient:

$$\hat{\nabla}_\phi \mathcal{L} = \frac{1}{S} \sum_s \nabla_\phi \log p(\mathcal{D} \mid w^{(s)}) - \nabla_\phi D_{\mathrm{KL}}(q_\phi \| p(w))$$

4. Update $\phi \leftarrow \phi + \eta \, \hat{\nabla}_\phi \mathcal{L}$ with learning rate η.

Output: Gaussian approximation $q^*(w) = \mathcal{N}(\mu^*, \Sigma^*)$ to the posterior $p(w \mid \mathcal{D})$.

Comparison with Langevin. In the previous chapter, we sampled from $p(w \mid \mathcal{D})$ directly. Here we find the best Gaussian approximation to it. The Langevin posterior mean and the VI posterior mean μ^* are both close to the MAP estimate; the difference is in the covariance. VI with a diagonal Gaussian ignores posterior correlations between coordinates; Langevin samples capture them. The cost of VI is $O(p)$ storage for $(\mu, \Sigma^{1/2})$ and $O(np)$ per gradient step; Langevin costs the same per step but requires thousands of steps and stores all samples.

10.2 Variational Inference: The General Framework

We now abstract from the logistic regression example to the general setting.

The Problem

Let θ be a latent quantity (parameter or latent variable) and $\mathcal{D}$ observed data. We want to compute the posterior $p(\theta \mid \mathcal{D})$ but cannot do so analytically. We choose a **variational family** $\mathcal{Q} = \{q(\theta;\phi) : \phi \in \Phi\}$ and find:

$$\phi^* = \arg\min_{\phi} D_{\mathrm{KL}}(q(\theta;\phi) \| p(\theta \mid \mathcal{D}))$$

Definition 10.1 (Evidence Lower Bound)**.** The **ELBO** is:

$$\mathcal{L}(\phi) = \mathbb{E}_{q_\phi}[\log p(\theta, \mathcal{D})] - \mathbb{E}_{q_\phi}[\log q(\theta;\phi)]$$

It satisfies the fundamental decomposition:

$$\log p(\mathcal{D}) = \mathcal{L}(\phi) + D_{\mathrm{KL}}(q_\phi \| p(\cdot \mid \mathcal{D}))$$

Proof.

$$\begin{aligned}\mathcal{L}(\phi) &= \mathbb{E}_{q_\phi}[\log p(\theta, \mathcal{D})] - \mathbb{E}_{q_\phi}[\log q(\theta;\phi)] \\ &= \mathbb{E}_{q_\phi}[\log p(\theta \mid \mathcal{D})] + \log p(\mathcal{D}) - \mathbb{E}_{q_\phi}[\log q(\theta;\phi)] \\ &= \log p(\mathcal{D}) - D_{\mathrm{KL}}(q_\phi \| p(\cdot \mid \mathcal{D}))\end{aligned}$$

Rearranging: $\log p(\mathcal{D}) = \mathcal{L}(\phi) + D_{\mathrm{KL}}(q_\phi \| p(\cdot \mid \mathcal{D}))$. □

Since $D_{\mathrm{KL}} \geq 0$, the ELBO is a lower bound on the log evidence: $\mathcal{L}(\phi) \leq \log p(\mathcal{D})$. Maximizing the ELBO is equivalent to minimizing $D_{\mathrm{KL}}(q_\phi \| p(\cdot \mid \mathcal{D}))$. We maximize the ELBO because it involves only the joint $p(\theta, \mathcal{D}) = p(\mathcal{D} \mid \theta)\, p(\theta)$, never the intractable normalizing constant.

The ELBO also decomposes as:

$$\mathcal{L}(\phi) = \underbrace{\mathbb{E}_{q_\phi}[\log p(\mathcal{D} \mid \theta)]}_{\text{fit to data}} - \underbrace{D_{\mathrm{KL}}(q_\phi \| p(\theta))}_{\text{distance from prior}}$$

Maximizing the ELBO negotiates the same tradeoff as Bayesian inference itself: fit the data well, but don't stray too far from the prior.

10.3 Origins in Statistical Physics: Mean Field Theory

Variational inference did not originate in statistics. Its roots lie in theoretical physics, specifically in mean field theory: a technique for approximating the behavior of interacting particle systems.

The Ising Model

The canonical example is the **Ising model**, a mathematical model of ferromagnetism. Consider n spins $s_i \in \{-1, +1\}$ arranged on a lattice. Each pair of neighboring spins interacts with energy $-Js_is_j$ (they prefer to align when $J > 0$), and each spin interacts with an external field h with energy $-hs_i$. The total energy is:

$$E(s) = -J \sum_{\langle i,j \rangle} s_i s_j - h \sum_i s_i$$

where $\langle i, j \rangle$ denotes neighboring pairs. The Boltzmann distribution assigns probability:

$$p(s) = \frac{1}{Z} e^{-E(s)/T} \propto \exp\left(\frac{J}{T} \sum_{\langle i,j \rangle} s_i s_j + \frac{h}{T} \sum_i s_i \right)$$

Computing the partition function $Z = \sum_s e^{-E(s)/T}$ requires summing over 2^n configurations: intractable for any realistic n.

Mean Field Approximation

Mean field theory approximates the joint distribution $p(s)$ with a factorized distribution:

$$q(s; m) = \prod_{i=1}^{n} q_i(s_i; m_i), \qquad q_i(s_i; m_i) = \frac{1 + m_i s_i}{2}$$

where $m_i = \mathbb{E}_{q_i}[s_i] \in [-1, 1]$ is the mean spin at site i. The approximation q assumes the spins are independent: each spin experiences only the *average* effect of its neighbors, not their actual fluctuating values. This is the "mean field" — each spin sees the field of the average.

The optimal m_i minimizes $D_{\mathrm{KL}}(q \| p)$, or equivalently maximizes the ELBO. The free energy (negative ELBO) is:

$$\begin{aligned} F(m) &= -\mathcal{L}(m) \\ &= -\mathbb{E}_q[\log p(s)] - H(q) \\ &= E_{\mathrm{MF}}(m) - TH(q) \end{aligned}$$

where $H(q) = -\sum_i \mathbb{E}_{q_i}[\log q_i]$ is the entropy of q and $E_{\mathrm{MF}}(m) = -J \sum_{\langle i,j \rangle} m_i m_j - h \sum_i m_i$ is the mean field energy (with actual spins replaced by their means). Setting $\partial F / \partial m_i = 0$ gives the **mean field equations**:

$$m_i = \tanh\left(\frac{1}{T} \left[J \sum_{j \sim i} m_j + h \right] \right)$$

Each m_i is determined self-consistently by the average field of its neighbors. These equations are solved iteratively: start with initial $m_i^{(0)}$, update each m_i in turn using the

current values of the others, and repeat until convergence. This is exactly coordinate ascent on the ELBO — the same algorithm as variational inference in statistics, applied to a physics problem.

Mean field VI in statistics is the physicist's mean field theory, reinterpreted. In physics: spins interact and the mean field replaces neighbor interactions with averages. In statistics: latent variables are correlated in the posterior, and the mean field (factorized) approximation replaces them with independent marginals. In both cases: the free energy (negative ELBO) is minimized by a set of self-consistency equations, solved by coordinate ascent.

Mean field VI in Bayesian statistics. The mean field approximation assumes the posterior factorizes over blocks of variables:

$$q(\theta; \phi) = \prod_{j=1}^{k} q_j(\theta_j; \phi_j)$$

For example, in the Gaussian mixture model of Chapter 9, the mean field approximation factorizes over the assignment variables z and the parameters (π, μ_1, μ_2):

$$q(z, \pi, \mu_1, \mu_2) = q(z)\, q(\pi)\, q(\mu_1)\, q(\mu_2)$$

The optimal q_j can be found analytically by coordinate ascent on the ELBO:

$$\log q_j^*(\theta_j) = \mathbb{E}_{q_{-j}}[\log p(\theta, \mathcal{D})] + C$$

where the expectation is over all blocks except j. For conjugate models, these updates are in the exponential family and have closed forms — mean field VI in conjugate models is algebraically similar to Gibbs sampling, but replaces random draws from conditionals with deterministic updates of the variational parameters.

10.4 KL Divergence: Two Directions, Two Behaviors

The KL divergence is not symmetric: $D_{\mathrm{KL}}(q\|p) \neq D_{\mathrm{KL}}(p\|q)$ in general. The two directions lead to fundamentally different optimization problems with different behaviors.

Definition 10.2 (KL Divergence)**.** For distributions p and q over the same space:

$$D_{\mathrm{KL}}(p\|q) = \mathbb{E}_p\left[\log \frac{p(\theta)}{q(\theta)}\right] = \int p(\theta) \log \frac{p(\theta)}{q(\theta)}\, d\theta$$

This is always ≥ 0, with equality iff $p = q$ a.e. It is not a metric: it is not symmetric and does not satisfy the triangle inequality.

Direction 1: $D_{\mathrm{KL}}(p_{\mathrm{data}} \| p_\theta)$ — Maximum Likelihood Estimation

Let p_{data} be the true data distribution and p_θ be a parametric model. Consider minimizing:

$$D_{\mathrm{KL}}(p_{\mathrm{data}} \| p_\theta) = \mathbb{E}_{p_{\mathrm{data}}}[\log p_{\mathrm{data}}(x) - \log p_\theta(x)] = C - \mathbb{E}_{p_{\mathrm{data}}}[\log p_\theta(x)]$$

Since C does not depend on θ, minimizing $D_{\mathrm{KL}}(p_{\mathrm{data}} \| p_\theta)$ over θ is equivalent to maximizing:

$$\mathbb{E}_{p_{\mathrm{data}}}[\log p_\theta(x)] \approx \frac{1}{n} \sum_{i=1}^{n} \log p_\theta(x_i)$$

This is exactly **maximum likelihood estimation**. The parameters θ appear in the *second* argument of the KL divergence.

The behavior of $D_{\mathrm{KL}}(p_{\mathrm{data}} \| p_\theta)$. This divergence penalizes p_θ for assigning low probability to regions where p_{data} is high: the integral $\int p_{\mathrm{data}}(x) \log p_{\mathrm{data}}(x)/p_\theta(x)\, dx$ blows up if $p_\theta(x) \approx 0$ while $p_{\mathrm{data}}(x) > 0$. Therefore, minimizing $D_{\mathrm{KL}}(p_{\mathrm{data}} \| p_\theta)$ forces p_θ to cover all modes of p_{data}. If p_{data} is multimodal and p_θ cannot represent all modes, p_θ spreads its mass to cover all of them: **mass-covering** (also called **zero-avoiding**).

Direction 2: $D_{\mathrm{KL}}(q_\phi \| p_{\mathrm{target}})$ — Variational Inference

In VI, the parameters ϕ appear in the *first* argument:

$$D_{\mathrm{KL}}(q_\phi \| p_{\mathrm{target}}) = \mathbb{E}_{q_\phi}\left[\log \frac{q_\phi(\theta)}{p_{\mathrm{target}}(\theta)}\right]$$

This divergence penalizes q_ϕ for assigning high probability to regions where p_{target} is low: the integral blows up if $q_\phi(\theta) > 0$ while $p_{\mathrm{target}}(\theta) \approx 0$. Therefore, minimizing $D_{\mathrm{KL}}(q_\phi \| p_{\mathrm{target}})$ forces q_ϕ to avoid putting mass where p_{target} is near zero. If p_{target} is multimodal and q_ϕ cannot cover all modes, q_ϕ concentrates on one mode rather than spreading: **mode-seeking** (also called **zero-forcing**).

The two KL divergences.

Direction	Name	Behavior
$D_{\mathrm{KL}}(p_{\mathrm{data}} \| p_\theta)$	MLE	Mass-covering p_θ covers all modes of p_{data}
$D_{\mathrm{KL}}(q_\phi \| p_{\mathrm{target}})$	VI	Mode-seeking q_ϕ picks one mode of p_{target}

The parameter being optimized (θ or ϕ) appears on different sides of the KL divergence. This structural difference leads to qualitatively different behaviors

in multimodal settings. Neither direction gives the correct posterior; they err in opposite directions.

Mode-seeking behavior is the fundamental limitation of VI with $D_{\mathrm{KL}}(q_\phi \| p_{\mathrm{target}})$. In a bimodal posterior (e.g., two symmetric modes from label switching in a mixture model), VI will typically select one mode and assign zero mass to the other. The resulting approximate posterior is wrong in a way that is hard to detect without comparison to a ground truth.

10.5 Variational Autoencoders

The ELBO is not only a tool for approximating posteriors over parameters. It is also the objective function for a class of generative models that use variational inference over *latent variables* at the level of individual data points. This leads to the **Variational Autoencoder (VAE)**, introduced by Kingma and Welling (2013), one of the foundational models of modern generative AI.

From Parameters to Latent Variables

In the previous sections, θ was a global parameter (e.g., the weight vector w in logistic regression): a single vector shared across all observations. Now we consider a different setup: each data point x_i has its own local latent variable z_i.

The generative model is:

$$\begin{aligned} z &\sim p(z) = \mathcal{N}(0, I_k) \\ x \mid z &\sim p_\theta(x \mid z) \end{aligned}$$

The prior over z is a standard Gaussian. The conditional $p_\theta(x \mid z)$ is a neural network with parameters θ (the **decoder**): given a latent code z, it generates an observation x. For image data, $p_\theta(x \mid z) = \mathcal{N}(\mu_\theta(z), I)$ where $\mu_\theta(z)$ is a neural network mapping z to the pixel mean.

The marginal likelihood for a single observation is:

$$p_\theta(x) = \int p_\theta(x \mid z)\, p(z)\, dz$$

This integral is intractable: the latent variable z runs over $\mathbb{R}^k$ and the integrand is a complex neural network function. We cannot compute $p_\theta(x)$ or its gradient with respect to θ.

The VAE ELBO

We introduce a variational approximation to the posterior $p_\theta(z \mid x)$: the **encoder** $q_\phi(z \mid x) = \mathcal{N}(z; \mu_\phi(x), \Sigma_\phi(x))$, where $\mu_\phi(x)$ and $\Sigma_\phi(x)$ are neural networks with parameters ϕ (typically diagonal covariance: $\Sigma_\phi(x) = \mathrm{diag}(\sigma^2_{\phi,1}(x), \ldots, \sigma^2_{\phi,k}(x))$).

The ELBO for a single data point x is:

$$\mathcal{L}(\theta, \phi; x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x \mid z)] - D_{\mathrm{KL}}(q_\phi(z \mid x) \| p(z))$$

The total ELBO over the dataset:

$$\mathcal{L}(\theta, \phi) = \sum_{i=1}^{n} \mathcal{L}(\theta, \phi; x_i) = \sum_{i=1}^{n} \left[\mathbb{E}_{q_\phi(z|x_i)}[\log p_\theta(x_i \mid z)] - D_{\mathrm{KL}}(q_\phi(z \mid x_i) \| p(z))\right]$$

The two terms have clean interpretations:

- **Reconstruction term:** $\mathbb{E}_{q_\phi(z|x_i)}[\log p_\theta(x_i \mid z)]$. Encode x_i to get a distribution over z, sample z, then decode back to x_i. Reward how well the decoded x_i matches the original. This is the **autoencoder** loss: how faithfully x_i is reconstructed.
- **Regularization term:** $-D_{\mathrm{KL}}(q_\phi(z \mid x_i) \| p(z))$. Penalize the encoder for mapping x_i to a distribution over z that is far from the prior $\mathcal{N}(0, I_k)$. This forces the latent codes to be organized: similar x_i should produce similar z_i, and the latent space should be "well-shaped" enough to allow generation by sampling $z \sim \mathcal{N}(0, I_k)$.

The KL term between two Gaussians is analytic:

$$D_{\mathrm{KL}}(q_\phi(z \mid x) \| p(z)) = \frac{1}{2} \sum_{j=1}^{k} \left[\sigma_{\phi,j}^2(x) + \mu_{\phi,j}^2(x) - 1 - \log \sigma_{\phi,j}^2(x)\right]$$

The reconstruction term is estimated by the reparameterization trick: sample $\epsilon \sim \mathcal{N}(0, I_k)$, form $z = \mu_\phi(x) + \Sigma_\phi^{1/2}(x)\, \epsilon$, and evaluate $\log p_\theta(x \mid z)$.

The VAE is simultaneous VI and generative modeling. The encoder $q_\phi(z \mid x)$ approximates the intractable posterior $p_\theta(z \mid x)$. The decoder $p_\theta(x \mid z)$ defines the generative model. Both are trained jointly by maximizing the ELBO with backpropagation through the reparameterization $z = \mu_\phi(x) + \Sigma_\phi^{1/2}(x)\, \epsilon$. After training:

- **Generation:** sample $z \sim \mathcal{N}(0, I)$, pass through decoder to get $x \sim p_\theta(x \mid z)$.
- **Inference:** encode x with $q_\phi(z \mid x)$ to get the posterior over latent codes.
- **Interpolation:** encode two data points to get z_1 and z_2, interpolate in latent space, decode.

Connection to the EM Algorithm

The **Expectation-Maximization (EM) algorithm** is a special case of VI where the variational distribution is taken to be the exact posterior.

In EM, we maximize $\log p_\theta(\mathcal{D}) = \sum_i \log p_\theta(x_i)$ over parameters θ when the model has latent variables z. The EM algorithm alternates:

E-step: compute the exact posterior over latent variables:

$$q^{(t)}(z \mid x_i) = p_{\theta^{(t)}}(z \mid x_i)$$

M-step: update parameters to maximize the ELBO with q fixed:

$$\theta^{(t+1)} = \arg\max_\theta \sum_i \mathbb{E}_{q^{(t)}(z|x_i)}[\log p_\theta(x_i, z)]$$

The ELBO at the E-step becomes tight ($D_{\mathrm{KL}} = 0$ since $q = p_\theta(z \mid x)$ exactly), so the ELBO equals $\log p_\theta(\mathcal{D})$. The M-step then increases $\log p_\theta(\mathcal{D})$ directly.

EM is VI with the exact posterior. The general VI objective is:

$$\mathcal{L}(\theta, \phi) = \sum_i \mathbb{E}_{q_\phi(z|x_i)}[\log p_\theta(x_i, z)] - \mathbb{E}_{q_\phi(z|x_i)}[\log q_\phi(z \mid x_i)]$$

- **EM:** fix $q_\phi(z \mid x_i) = p_{\theta^{(t)}}(z \mid x_i)$ (exact posterior). Update θ by M-step. Exact but requires a tractable posterior.
- **VAE:** parameterize $q_\phi(z \mid x_i)$ as a neural network encoder. Update both ϕ and θ jointly by gradient ascent. Approximate but applies to any differentiable generative model.

When the exact posterior is tractable (e.g., Gaussian mixture models), EM is preferred. When it is not (deep generative models), the VAE's amortized inference encoder $q_\phi(z \mid x)$ provides a scalable approximation.

10.6 Diffusion Models: Score Functions and ELBO

We now develop diffusion models as a synthesis of the ideas in this chapter: the score function from Chapter 9 (Langevin dynamics), the ELBO from VAE, and the physical/mental directions from Chapter 2. Diffusion models are among the most powerful generative models in modern AI; their mathematical foundations are precisely the tools we have been building.

The Physical Direction: Forward Noising Process

Let $x_0 \sim p_{\mathrm{data}}(x)$ be a data sample (e.g., an image). The forward process adds Gaussian noise over T steps:

$$x_t = \sqrt{1 - \beta_t}\, x_{t-1} + \sqrt{\beta_t}\, \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, I), \quad t = 1, \ldots, T$$

where $\beta_1, \ldots, \beta_T$ is a noise schedule with $0 < \beta_t < 1$. This is the **physical direction**: a known, simple process that progressively destroys the structure of x_0. After T steps, $x_T \approx \mathcal{N}(0, I)$ regardless of x_0: all information has been erased.

Define $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$. A key property of this process: the marginal at any time t given x_0 is:

$$x_t \mid x_0 \;\sim\; \mathcal{N}(\sqrt{\bar{\alpha}_t}\, x_0,\; (1-\bar{\alpha}_t)\, I)$$

This allows sampling x_t directly from x_0 without iterating t steps.

The Mental Direction: Deriving the Backward Step

To generate new data, we need to run the process backward: starting from noise x_T, recover a data sample x_0. This is the **mental direction**: reasoning from effect (x_T, noisy) back to cause (x_0, clean).

The exact backward step is:

$$p(x_{t-1} \mid x_t) \propto p(x_{t-1})\, p(x_t \mid x_{t-1})$$

We derive the backward distribution analytically using a first-order Taylor expansion of $\log p(x_{t-1})$ in a neighborhood of x_t.

Derivation via First-Order Taylor Expansion

We work in the continuous-time limit with small step size $\beta \ll 1$ (dropping subscripts for clarity). The forward step is:

$$x_t = \sqrt{1-\beta}\, x_{t-1} + \sqrt{\beta}\, \varepsilon \approx x_{t-1} + \sqrt{\beta}\, \varepsilon - \frac{\beta}{2}\, x_{t-1}$$

For small β, x_{t-1} is close to x_t. The forward kernel is:

$$p(x_t \mid x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta}\, x_{t-1},\, \beta I)$$

The log of the unnormalized backward is:

$$\begin{aligned} \log p(x_{t-1} \mid x_t) &= \log p(x_{t-1}) + \log p(x_t \mid x_{t-1}) + C \\ &= \log p(x_{t-1}) - \frac{\|x_t - \sqrt{1-\beta}\, x_{t-1}\|^2}{2\beta} + C \end{aligned}$$

Since x_{t-1} is close to x_t, expand $\log p(x_{t-1})$ to first order around x_t:

$$\log p(x_{t-1}) \approx \log p(x_t) + (x_{t-1} - x_t)^\top \nabla \log p(x_t)$$

Let $u = x_{t-1} - x_t$ (the displacement from x_t to x_{t-1}). The exponent becomes a quadratic in u:

$$(x_t + u)^\top \nabla \log p(x_t) - \frac{\|x_t - \sqrt{1-\beta}(x_t + u)\|^2}{2\beta} + C$$

$$= u^\top \nabla \log p(x_t) - \frac{\|\beta x_t/2 - \sqrt{1-\beta}\, u\|^2}{2\beta} + C' \qquad (\text{using } \sqrt{1-\beta} \approx 1 - \beta/2)$$

$$\approx u^\top \nabla \log p(x_t) - \frac{\|u\|^2}{2\beta} + C''$$

Completing the square in u:

$$= -\frac{1}{2\beta}\|u - \beta \nabla \log p(x_t)\|^2 + C'''$$

Therefore:

$$x_{t-1} - x_t \sim \mathcal{N}(\beta \nabla \log p(x_t),\ \beta I)$$

equivalently:

Backward step (score-based):

$$p(x_{t-1} \mid x_t) \approx \mathcal{N}(x_{t-1};\ x_t + \beta \nabla \log p_t(x_t),\ \beta I)$$

The backward step is Gaussian with mean $x_t + \beta \nabla \log p_t(x_t)$ and variance βI. The **score function** $\nabla \log p_t(x_t)$ is the correction that steers the backward step toward higher density regions. Without it, the backward step would be pure noise.

This is the Langevin update in disguise: the backward step is one step of Langevin dynamics on the density p_t, with step size β and the standard noise $\sqrt{\beta}\,\xi$ (absorbed into the Gaussian form above). The connection between diffusion models and Langevin dynamics is exact: both use the score function to navigate toward high-probability regions of a density.

Learning the Score

The score $\nabla \log p_t(x_t)$ is unknown — it is the gradient of the log density of the noisy distribution at time t. We learn it with a neural network $s_\theta(x, t) \approx \nabla \log p_t(x)$.

The key insight: since $x_t \mid x_0 \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}\, x_0,\ (1 - \bar{\alpha}_t)I)$, the score can be related to the noise ε that was added:

$$\nabla_{x_t} \log p_t(x_t \mid x_0) = -\frac{x_t - \sqrt{\bar{\alpha}_t}\, x_0}{1 - \bar{\alpha}_t} = -\frac{\varepsilon}{\sqrt{1 - \bar{\alpha}_t}}$$

This motivates learning a **noise prediction network** $\varepsilon_\theta(x_t, t)$ that predicts the noise added at step t. The score is then:

$$s_\theta(x_t, t) \approx -\frac{\varepsilon_\theta(x_t, t)}{\sqrt{1 - \bar{\alpha}_t}}$$

The training objective is:

$$\begin{aligned} L_{\text{simple}}(\theta) &= \mathbb{E}_{t,x_0,\varepsilon}\big[\|\varepsilon - \varepsilon_\theta(x_t, t)\|^2\big] \\ &= \mathbb{E}_{t,x_0,\varepsilon}\big[\|\varepsilon - \varepsilon_\theta(\sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \varepsilon, t)\|^2\big] \end{aligned}$$

This is a denoising objective: given a noisy version x_t of a clean data point x_0, predict the noise ε that was added. The network is trained on pairs (x_t, ε) generated by the forward process.

The VAE View of Diffusion Models

The ELBO provides a principled objective for training diffusion models, connecting them directly to VAEs. Treat the forward process as a fixed encoder (the variational distribution q) and the backward process as a learned decoder (p_θ).

Forward process as encoder:

$$q(x_{1:T} \mid x_0) = \prod_{t=1}^{T} q(x_t \mid x_{t-1})$$

This is a fixed (non-learned) Markov chain that encodes x_0 into a sequence of progressively noisier versions.

Backward process as decoder:

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t)$$

This is a learned Markov chain that starts from $p(x_T) = \mathcal{N}(0, I)$ and generates data by running the learned backward steps.

The ELBO for a single data point x_0 is:

$$\begin{aligned} \mathcal{L}(\theta; x_0) &= \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{p_\theta(x_{0:T})}{q(x_{1:T} \mid x_0)}\right] \\ &= \underbrace{\mathbb{E}_{q(x_1|x_0)}[\log p_\theta(x_0 \mid x_1)]}_{\text{reconstruction}} - \underbrace{D_{\text{KL}}(q(x_T \mid x_0) \| p(x_T))}_{\text{prior matching}} \\ &\quad - \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q(x_t|x_0)}[D_{\text{KL}}(q(x_{t-1} \mid x_t, x_0) \| p_\theta(x_{t-1} \mid x_t))]}_{\text{denoising step } t} \end{aligned}$$

The three terms are:

- **Reconstruction:** how well p_θ recovers x_0 from x_1 (mildly noisy version).
- **Prior matching:** how close the fully noised x_T is to the prior $\mathcal{N}(0, I)$. With enough noise steps and a suitable schedule, this is approximately zero.

- **Denoising steps:** at each step t, how well the learned backward $p_\theta(x_{t-1} \mid x_t)$ matches the true backward $q(x_{t-1} \mid x_t, x_0)$. This is the dominant term.

The key insight: the true backward kernel $q(x_{t-1} \mid x_t, x_0)$ is tractable given x_0:

$$q(x_{t-1} \mid x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0),\ \tilde{\beta}_t I)$$

where:

$$\tilde{\mu}_t(x_t, x_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\,\beta_t}{1-\bar{\alpha}_t}\,x_0 + \frac{\sqrt{1-\beta_t}\,(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\,x_t, \qquad \tilde{\beta}_t = \frac{(1-\bar{\alpha}_{t-1})\,\beta_t}{1-\bar{\alpha}_t}$$

This is derived by applying Bayes rule to the Gaussian forward process: the backward step given both x_t and x_0 is Gaussian. The denoising term then becomes:

$$D_{\mathrm{KL}}(q(x_{t-1} \mid x_t, x_0) \| p_\theta(x_{t-1} \mid x_t)) = \frac{1}{2\tilde{\beta}_t}\|\tilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t)\|^2 + C$$

Training minimizes this KL at each step: the learned backward mean $\mu_\theta(x_t, t)$ should match the true backward mean $\tilde{\mu}_t(x_t, x_0)$. Substituting the expression for $\tilde{\mu}_t$ in terms of $x_0 = (x_t - \sqrt{1-\bar{\alpha}_t}\,\varepsilon)/\sqrt{\bar{\alpha}_t}$, the objective reduces to the simple noise-prediction loss $\|\varepsilon - \varepsilon_\theta(x_t, t)\|^2$.

Diffusion models are hierarchical VAEs.

- The forward process (encoder) is fixed and Gaussian: it maps data to noise through T steps.
- The backward process (decoder) is learned: it maps noise to data through T denoising steps.
- The training objective is the ELBO, which reduces to predicting the noise at each step.
- The score function $\nabla \log p_t(x_t)$, which Langevin dynamics uses for sampling, is proportional to the negative noise: $-\varepsilon_\theta(x_t, t)/\sqrt{1-\bar{\alpha}_t}$.

The connection between Langevin dynamics (sampling via score) and diffusion models (generation via learned score) is exact: both use the same mathematical object (the score function) for the same purpose (navigating toward high-probability regions of a density). The difference is that Langevin dynamics uses the known score of a fixed target, while diffusion models learn the score of the unknown data distribution.

Exercise 10.3. Show that the ELBO for the VAE can be written as:

$$\mathcal{L}(\theta, \phi; x) = \mathbb{E}_{\epsilon \sim \mathcal{N}(0,I)}\Big[\log p_\theta(x \mid \mu_\phi(x) + \Sigma_\phi^{1/2}(x)\,\epsilon)\Big] - D_{\mathrm{KL}}(q_\phi(z \mid x) \| p(z))$$

Explain why the reparameterization is necessary for backpropagation through the expectation over z.

Exercise 10.4. Consider a Gaussian target $p(\theta) = \mathcal{N}(\mu, \Sigma)$ with two modes: specifically a mixture $p(\theta) = \frac{1}{2}\mathcal{N}(\mu_1, I) + \frac{1}{2}\mathcal{N}(\mu_2, I)$ with $\|\mu_1 - \mu_2\| = 10$. Let $q_\phi(\theta) = \mathcal{N}(\theta; m, s^2 I)$.

1. Explain qualitatively what the optimal q_ϕ looks like under $D_{\mathrm{KL}}(q_\phi \| p)$. Which mode does it select and why?

2. Explain what the optimal q_ϕ looks like under $D_{\mathrm{KL}}(p \| q_\phi)$. Does it cover both modes? Why?

3. What is the fundamental reason for this asymmetry? Which term in each KL divergence is responsible for the different behaviors?

Exercise 10.5. In the VAE, the reconstruction loss for Gaussian output $p_\theta(x \mid z) = \mathcal{N}(\mu_\theta(z), I)$ is:

$$\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x \mid z)] \approx -\frac{1}{2}\|x - \mu_\theta(z^{(s)})\|^2 + C$$

for a single sample $z^{(s)} \sim q_\phi(z \mid x)$. Show that maximizing the ELBO is therefore equivalent to minimizing a sum of a reconstruction loss and a KL regularization term, and explain the role of each term in preventing the VAE from collapsing to a trivial solution (either ignoring the latent code or ignoring the data).

Exercise 10.6. In a diffusion model with $T = 1000$ steps and noise schedule $\beta_t = 0.02$ for all t:

1. Compute $\bar{\alpha}_T = \prod_{t=1}^{T}(1 - \beta_t) = (0.98)^{1000}$. Is x_T approximately Gaussian?

2. At step $t = 500$, compute $\sqrt{\bar{\alpha}_{500}}$ and $\sqrt{1 - \bar{\alpha}_{500}}$. What fraction of x_{500} is signal versus noise?

3. Show that the backward step formula $p(x_{t-1} \mid x_t) \approx \mathcal{N}(x_t + \beta \nabla \log p_t(x_t), \beta I)$ is exactly a Langevin step on the density p_t with step size β.

10.7 Latent Variable Models: Bayes vs. Bayesian

The VAE of the previous section is built on a **latent variable model**:

$$z \sim p(z), \qquad x \mid z \sim p_\theta(x \mid z)$$

The latent variable z represents hidden structure underlying the observed data x: the style behind a sentence, the object behind an image, the cause behind a sensory signal. The parameter θ — the weights of the decoder network — defines the generative model. This two-level structure is the foundation of an enormous range of models in statistics, machine learning, and cognitive science.

It is also the setting where a conceptually important distinction is most clearly drawn: the distinction between being **Bayes** and being **Bayesian**.

Being Bayes: Inferring z from x

Given a fixed, trained parameter θ, the posterior over the latent variable for a single observation x is:

$$p_\theta(z \mid x) = \frac{p_\theta(x \mid z)\, p(z)}{p_\theta(x)}$$

Computing or approximating this posterior is **using Bayes rule**. The prior $p(z)$ encodes beliefs about the latent structure before seeing x; the likelihood $p_\theta(x \mid z)$ encodes how x is generated from z; the posterior $p_\theta(z \mid x)$ encodes the inferred latent structure given the observation.

This is what the VAE encoder does: it approximates $p_\theta(z \mid x)$ with $q_\phi(z \mid x)$. This is what MCMC does when applied to the latent variable: it samples from $p_\theta(z \mid x)$. In both cases, the inference is about z, not about θ. The parameter θ is treated as fixed and known.

This is the Bayes position: apply Bayes rule to infer the latent variable z from the observation x. It is the mental direction, applied to hidden causes. Most practitioners of latent variable models occupy this position. The prior $p(z)$ on the latent variable is standard and uncontroversial; no one debates whether the prior on z makes sense, because z is genuinely random — it varies across data points, it is drawn anew for each observation, and the prior $p(z)$ is its marginal distribution.

Being Bayesian: Also Inferring θ from All x

The **Bayesian position** goes further. It insists that θ is also unknown, and that we should place a prior on θ and infer it from all the observations $x_1, \ldots, x_n$:

$$p(\theta) \qquad \text{(prior on parameters)}$$

$$\theta \mid x_1, \ldots, x_n \;\sim\; p(\theta \mid x_1, \ldots, x_n) \propto p(\theta) \prod_{i=1}^{n} p_\theta(x_i)$$

where $p_\theta(x_i) = \int p_\theta(x_i \mid z)\, p(z)\, dz$ is the marginal likelihood of each observation. The full Bayesian treatment requires inference over both θ and z: the joint posterior is:

$$p(\theta, z_{1:n} \mid x_{1:n}) \propto p(\theta) \prod_{i=1}^{n} p(z_i)\, p_\theta(x_i \mid z_i)$$

This is a much harder computational problem. θ is shared across all observations; z_i is local to observation i. Integrating over both requires either MCMC (expensive for large n and high-dimensional θ) or a structured variational approximation.

The Bayes/Bayesian distinction in latent variable models.

	Bayes	**Bayesian**
Unknown	Latent variable z	Latent variable z and parameter θ
Prior on	z: the prior $p(z)$	z and θ: priors $p(z)$ and $p(\theta)$
Inference	$p_\theta(z \mid x)$ for each x, with θ fixed	$p(\theta, z_{1:n} \mid x_{1:n})$ jointly
θ treatment	Estimated by MLE or MAP, then fixed	Random variable with a posterior
Example	VAE encoder	Bayesian VAE, full hierarchical model
Who does this	Most practitioners	Few practitioners

Why most people are Bayes but not Bayesian. In practice, the parameter θ of a deep generative model has millions or billions of dimensions. Maintaining a posterior over θ — even an approximate one — is computationally daunting. The VAE trains θ by maximizing the ELBO, which is point estimation of θ (MAP if a prior on θ is included, MLE if not). After training, θ is fixed, and only the inference over z per observation is Bayesian.

This is a pragmatic compromise, not a principled one. A fully Bayesian treatment would be more honest: it would acknowledge that θ is not known with certainty, would propagate uncertainty about θ into predictions, and would automatically avoid overfitting to the training data. But the computational cost is high, and for modern deep learning models it remains largely out of reach.

The distinction matters for generalization. When θ is estimated by MLE, the model can overfit to the training data: the learned θ may capture idiosyncrasies of $x_1, \ldots, x_n$ that do not generalize. A Bayesian posterior over θ, by contrast, integrates out parameter uncertainty and — as shown in the previous chapter — always produces better-calibrated predictions. The fully Bayesian latent variable model does not overfit; the MLE-trained VAE can.

The EM algorithm as the boundary case. The EM algorithm (Chapter 12) sits exactly on the boundary. It treats θ as a point estimate to be optimized (M-step) and z as a random variable to be marginalized (E-step). It is Bayes about z and frequentist about θ. This is the most common position in the literature, and it is more tractable than full Bayes while being more principled than ignoring z entirely.

The Brain as a Latent Variable Model

The latent variable model $p(z)\, p_\theta(x \mid z)$ is not only a statistical construction. It is, arguably, a model of perception itself.

The brain receives sensory signals x: photons striking the retina, pressure waves in the cochlea, chemical gradients on the tongue. These signals are noisy, ambiguous, and incomplete. From them, the brain must infer the structure of the external world: the objects, their locations, their identities, their relationships. The external world is the latent variable z; the sensory signals are the observation x.

This picture was articulated most influentially by the nineteenth-century physicist and physiologist **Hermann von Helmholtz** (1821–1894). Helmholtz argued that perception is not a passive recording of sensory input but an active process of inference. The brain, he proposed, unconsciously constructs the most plausible explanation for its sensory signals: it infers the hidden causes of what it sees, hears, and feels. Perception, in Helmholtz's account, is **unconscious inference**: the brain solves the inverse problem of recovering z from x.

The Helmholtz principle. Perception is the brain's best guess at the hidden causes of sensory signals. The brain does not experience the sensory signals directly; it experiences its inference about what caused them. What we call seeing, hearing, and feeling is the posterior $p(z \mid x)$, not the likelihood $p(x \mid z)$. The world is the latent variable; the senses are the noisy observation.

This is Bayes rule applied to perception. The prior $p(z)$ encodes the brain's expectations about the world before any sensory input arrives: the regularities of natural scenes, the typical shapes of objects, the common causes of sounds. The likelihood $p_\theta(x \mid z)$ encodes the generative model: how the world produces sensory signals. The posterior $p_\theta(z \mid x)$ is the percept: the brain's best estimate of the world given what the senses report.

The Hierarchy of Representations

The latent variable z in the brain is not a single layer. Neuroscience and machine learning both suggest a **hierarchical** structure: multiple layers of representation $z^{(1)}, z^{(2)}, \ldots, z^{(L)}$, each more abstract than the last, each encoding a different level of structure in the data.

$$p(z^{(L)})\, p(z^{(L-1)} \mid z^{(L)}) \cdots p(z^{(1)} \mid z^{(2)})\, p_\theta(x \mid z^{(1)})$$

In the visual cortex, early layers encode edges and orientations; intermediate layers encode shapes and textures; higher layers encode objects, faces, and scenes. Each layer is a latent variable that conditions on the layer above and generates the layer below. The inference problem is to recover all layers $z^{(1)}, \ldots, z^{(L)}$ from the raw sensory input x.

This hierarchical generative model is exactly the architecture of a deep VAE or a diffusion model. The decoder $p_\theta(x \mid z^{(1)})$ generates sensory data from low-level representations; the higher-level priors $p(z^{(\ell)} \mid z^{(\ell+1)})$ encode structural regularities at

each level of abstraction. The encoder $q_\phi(z \mid x)$ approximates the posterior inference: given the sensory input, infer the representations at every level.

The brain as a deep VAE. If the brain implements Helmholtz's principle in a hierarchical generative model, then:

- **Perception** is the approximate posterior inference $q_\phi(z \mid x) \approx p_\theta(z \mid x)$: recovering representations from sensory signals.
- **Imagination and dreaming** are samples from the prior: $z \sim p(z)$, $x \sim p_\theta(x \mid z)$, with no sensory input anchoring the inference.
- **Learning** is updating the generative model parameters θ: adjusting the brain's model of how the world generates sensory signals.
- **Attention** is directing inference resources toward the most informative or surprising parts of the sensory input: focusing the posterior update where it matters most.
- **Prediction error** is the residual $x - \mathbb{E}[x \mid z]$: the part of the sensory signal not explained by the current representation. In predictive coding theories, the brain transmits only prediction errors, not raw sensory signals, between layers.

Predictive coding. The neuroscientist Karl Friston has developed this picture into a comprehensive theory of brain function called **predictive coding** (or the free energy principle). In this framework, the brain continuously generates predictions of its sensory input from the current representation z, and updates z based on the prediction error. The brain is not a passive receiver of sensory signals but an active prediction machine, constantly testing its generative model against incoming data. The update rule is gradient descent on the surprise (negative log likelihood) of the sensory input: a continuous, online version of the variational inference ELBO maximization we developed in this chapter.

The free energy principle proposes that *all* brain processes — perception, action, learning, attention, sleep — can be understood as minimizing a variational free energy, which is the negative ELBO of the brain's generative model. This is an ambitious claim that remains debated, but it illustrates the reach of the variational inference framework beyond its origins in statistics and machine learning.

Is the brain Bayes or Bayesian? In the Helmholtz/predictive coding picture, the brain is clearly Bayes: it uses Bayes rule to infer z from x, with the generative model $p_\theta(x \mid z)$ fixed (or slowly adapting during learning). Whether the brain is also Bayesian — maintaining a posterior over the generative model parameters θ themselves — is an open question. During development and learning, the brain updates θ in response to experience: this is learning. But whether this update is a proper Bayesian posterior

update or a gradient-descent-like point estimation is not known.

Exercise 10.7. Consider a simple generative model for visual perception: $z \in \{\text{cat}, \text{dog}\}$ with $p(z = \text{cat}) = 0.4$, and noisy observations $x \mid z$ where $p(x = \text{fuzzy} \mid z = \text{cat}) = 0.8$ and $p(x = \text{fuzzy} \mid z = \text{dog}) = 0.3$.

1. Compute the posterior $p(z \mid x = \text{fuzzy})$. What is the brain's best guess at the hidden cause?

2. The observation is ambiguous: $x = \text{fuzzy}$ is consistent with both causes. How does the prior $p(z)$ resolve the ambiguity? What would happen if the prior were $p(z = \text{cat}) = 0.01$ (cats are very rare)?

3. In Helmholtz's framework, what does the posterior mean phenomenologically — what does the person "see"?

4. Suppose the person is very hungry and expects food, so their prior is now $p(z = \text{cat}) = 0.9$ (they want to see the cat so they can pet it and feel calm). How does the posterior change? What does this model about top-down influences on perception?

Exercise 10.8. The brain's generative model can be written as a deep hierarchical model:

$$p(z^{(2)})\, p(z^{(1)} \mid z^{(2)})\, p(x \mid z^{(1)})$$

with two layers of latent variables. Suppose all distributions are Gaussian: $z^{(2)} \sim \mathcal{N}(0, I)$, $z^{(1)} \mid z^{(2)} \sim \mathcal{N}(Wz^{(2)}, I)$, $x \mid z^{(1)} \sim \mathcal{N}(z^{(1)}, I)$.

1. Show that the marginal $p(x) = \mathcal{N}(0, WW^\top + 2I)$. What does the matrix W represent in the context of perceptual representations?

2. Write down the ELBO for this model with a factorized Gaussian variational family $q(z^{(1)}, z^{(2)} \mid x) = q(z^{(1)} \mid x)\, q(z^{(2)} \mid x)$. Identify the reconstruction term and the two KL regularization terms.

3. In the predictive coding interpretation, what is the prediction error at each layer? Show that minimizing the ELBO is equivalent to minimizing the sum of squared prediction errors across all layers.

Chapter 11

Large Language Models and Bayesian Prediction

The preceding chapters developed Bayesian inference as a framework for learning from data: a prior over unknown parameters, a likelihood connecting parameters to observations, and a posterior combining both. We have seen this framework applied to physical constants, regression coefficients, cluster assignments, and function spaces. This chapter argues that the same framework — in a precise mathematical sense — describes what happens inside a large language model when it generates text.

The argument has two parts. First, a large language model is trained to approximate a specific conditional distribution: the probability of the next token given all previous tokens. Second, this approximation, when done well enough, implicitly performs Bayesian prediction: the model infers latent structure from the context and integrates over that structure to predict what comes next. The model does not do this explicitly — there is no posterior computation inside a transformer. But the distribution it learns to approximate is exactly the distribution a Bayesian reasoner would produce. In this sense, a sufficiently powerful language model is a Bayesian predictor, with the implicit parameters integrated out.

11.1 Language Models as Autoregressive Distributions

The Basic Setup

A piece of text is a sequence of **tokens**: words, subwords, or characters drawn from a fixed vocabulary $\mathcal{V}$ of size $|\mathcal{V}|$ (typically 50,000–100,000 tokens for modern models). A document of length T is a sequence $x_1, x_2, \ldots, x_T$ with each $x_t \in \mathcal{V}$.

By the chain rule of probability, the joint distribution over any sequence factorizes

exactly as:

$$p(x_1, x_2, \ldots, x_T) = \prod_{t=1}^{T} p(x_t \mid x_1, \ldots, x_{t-1}) = \prod_{t=1}^{T} p(x_t \mid x_{<t})$$

This is not an approximation or a modeling assumption. It is the chain rule applied to the joint distribution, in the same way we derived the general chain rule for Bayes networks in Chapter 4. A language model is a parametric approximation to each factor:

$$p_\theta(x_t \mid x_{<t}) \approx p(x_t \mid x_{<t})$$

where θ denotes all learnable parameters. Generating text means sampling autoregressively: sample $x_1 \sim p_\theta(x_1)$, then $x_2 \sim p_\theta(x_2 \mid x_1)$, then $x_3 \sim p_\theta(x_3 \mid x_1, x_2)$, and so on. Each step conditions on everything generated so far.

Training. The model is trained on a large corpus of text by maximum likelihood: minimize the negative log-likelihood of the training data,

$$\mathcal{L}(\theta) = -\sum_{t=1}^{T} \log p_\theta(x_t \mid x_{<t})$$

summed over all documents in the training corpus. At each position t, the model sees the preceding tokens and must predict the next one. The training objective is exactly cross-entropy: the average number of bits needed to encode each token, given the context, under the model's distribution.

A language model is a single object: the conditional distribution $p(x_t \mid x_{<t})$. Everything else — question answering, translation, reasoning, code generation — is a consequence of approximating this distribution well over a sufficiently large and diverse corpus. The model learns to predict the next token; the emergent behaviors follow.

11.2 The Transformer: Embeddings and Learned Matrices

The transformer architecture (Vaswani et al., 2017) is the neural network used to implement $p_\theta(x_t \mid x_{<t})$ in all modern large language models. We sketch its key components at the level of abstraction needed for the Bayesian argument.

Token Embeddings

Each token $x_t \in \mathcal{V}$ is mapped to a vector $e_t \in \mathbb{R}^d$ by a learned **embedding matrix** $E \in \mathbb{R}^{|\mathcal{V}| \times d}$: the row of E corresponding to token x_t is the embedding e_t. The dimension d is the **model dimension** (typically 768 to 12,288 for modern models). The

embedding maps a discrete token into a continuous vector where semantic and syntactic relationships can be represented geometrically: similar tokens have similar embeddings, and relationships like "king - man + woman ≈ queen" emerge from training.

A position encoding $\text{pos}(t)$ is added to each embedding to give the model information about the position of each token in the sequence: $h_t^{(0)} = e_t + \text{pos}(t)$.

Attention: Reading the Context

The core operation of the transformer is **self-attention**. At each layer ℓ, each position t computes three vectors from its current representation $h_t^{(\ell)}$:

$$Q_t = W_Q h_t^{(\ell)}, \quad K_t = W_K h_t^{(\ell)}, \quad V_t = W_V h_t^{(\ell)}$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d \times d}$ are learned **query, key, and value matrices**. The attention score of position t to position s is:

$$a_{ts} = \frac{\exp(Q_t^\top K_s / \sqrt{d})}{\sum_{s' \le t} \exp(Q_t^\top K_{s'} / \sqrt{d})}$$

(Causal masking ensures that position t can only attend to positions $s \le t$, so the model cannot see future tokens.) The updated representation is:

$$h_t^{(\ell+1)} = \sum_{s \le t} a_{ts} V_s + \text{feedforward}(h_t^{(\ell)})$$

The attention weights a_{ts} measure how relevant position s is to predicting what comes after position t. The key, query, and value matrices are learned during training to make this relevance computation useful for next-token prediction.

The Output Distribution

After L layers of attention and feedforward operations, the final representation $h_t^{(L)} \in \mathbb{R}^d$ is projected back to the vocabulary by an **unembedding matrix** $U \in \mathbb{R}^{d \times |\mathcal{V}|}$:

$$p_\theta(x_{t+1} = v \mid x_{\le t}) = \text{softmax}(U^\top h_t^{(L)})_v = \frac{\exp(u_v^\top h_t^{(L)})}{\sum_{v'} \exp(u_{v'}^\top h_t^{(L)})}$$

where u_v is the column of U corresponding to token v. The model outputs a probability distribution over the entire vocabulary at each position.

Scale. A large language model has hundreds of layers, thousands of attention heads, and tens to hundreds of billions of parameters. GPT-3 has 175 billion parameters; GPT-4 is estimated to be larger still. The matrices $W_Q, W_K, W_V, W_O, W_1, W_2$ (attention and feedforward weights) at each layer, the embedding matrix E, and the unembedding matrix U together constitute θ.

Next Token Prediction as Continuous Rules

What does a trained language model actually do? One way to think about it: the learned matrices implement a form of **soft rule application**.

Consider a simple example. The phrase "the capital of France is" should be followed by "Paris." A lookup table could encode this rule exactly. But a language model encodes it in a different form: the embedding of "France" and "capital" interact through the attention mechanism to produce a representation $h_t^{(L)}$ that, when multiplied by the unembedding matrix U, gives high probability to the token "Paris."

This is not a discrete lookup. It is a continuous, distributed computation in which the matrices W_Q, W_K, W_V, E, U have been optimized over billions of examples to implement such associations as differentiable operations on vectors. The rules are not hard-coded; they are **soft patterns** encoded in the geometry of the embedding space and the learned matrices. A new phrase "the capital of Germany is" activates a similar but distinct geometric pattern and produces "Berlin."

A language model is a continuous, differentiable lookup system. The discrete rules of language — grammar, factual associations, stylistic conventions, logical inference — are represented implicitly as geometric relationships in the embedding space and as linear transformations in the learned matrices. The model does not execute rules; it interpolates among the patterns it has seen, with the interpolation implemented by matrix multiplication and attention.

What is Learned? Memorization and Generalization

A natural question is whether a language model merely memorizes its training data or generalizes beyond it. The answer is both, in different regimes.

Memorization. For specific facts, names, dates, and verbatim passages that appear frequently in training data, the model memorizes them directly: the weights encode the association so strongly that the model reproduces the training content. This is the direct analog of overfitting in regression.

Generalization. For patterns that appear in many forms across the training data — grammatical structure, reasoning patterns, coding conventions, mathematical identities — the model learns the underlying rule rather than specific instances. It can apply the rule to new inputs it has never seen. This is the analog of a well-regularized Bayesian posterior: the model has inferred the latent structure from many observations and can generalize to new ones.

The distinction matters for understanding what the model has learned. The memorized content is distributed across the weight matrices; the generalized patterns are distributed across the attention heads and layers. Both contribute to the next-token distribution, and both are invisibly blended in the output.

11.3 In-Context Learning: GPT-3's Surprise

The Discovery

When OpenAI released GPT-3 in 2020, the paper (Brown et al.) contained a result that surprised the research community. GPT-3 had been trained purely as a next-token predictor, with no explicit training for any downstream task. Yet it could solve arithmetic problems, translate languages, answer factual questions, and perform many other tasks — simply by including a few examples in the input prompt, without any change to the model's weights.

This phenomenon was called **in-context learning (ICL)**: the ability of a language model to adapt to a new task using only examples provided in the input context, with no gradient updates. The model was never trained to do this. It emerged as a consequence of training on enough data with a next-token objective.

The standard demonstration format is:

```
Input:  2 + 3   Output:  5
Input:  7 + 1   Output:  8
Input:  4 + 6   Output:  ?
```

Without any training on arithmetic, GPT-3 completes the sequence with `10`. It has inferred the rule from the examples in the context and applied it to the query. This works not only for arithmetic but for tasks described by natural language, demonstrated by examples, or specified by combinations of both.

A Simple Formal Setting: Bayesian Regression in Context

The Bayesian interpretation of in-context learning was articulated by Xie et al. (2021) and elaborated by several subsequent papers. We develop it in the simplest possible setting: linear regression with an unknown slope.

Consider a sequence of (x, y) pairs where $y = \beta x + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ and unknown slope β. The context is:

$$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n), (x_{n+1}, ?)$$

The task: predict y_{n+1} given the context.

The Bayesian answer. Place a prior $\beta \sim \mathcal{N}(0, \tau^2)$ on the unknown slope. From the Gaussian-Gaussian conjugate update of Chapter 5, the posterior after observing $(x_1, y_1), \ldots, (x_n, y_n)$ is:

$$\beta \mid x_{1:n}, y_{1:n} \;\sim\; \mathcal{N}(\hat{\beta}_n, \sigma_n^2)$$

where:

$$\hat{\beta}_n = \frac{\sum_i x_i y_i / \sigma^2}{1/\tau^2 + \sum_i x_i^2/\sigma^2}, \quad \frac{1}{\sigma_n^2} = \frac{1}{\tau^2} + \frac{\sum_i x_i^2}{\sigma^2}$$

The posterior predictive distribution for the next output is:

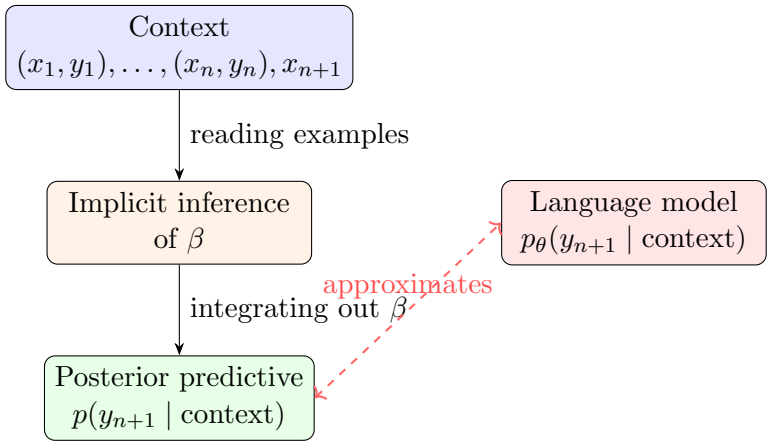
$$\begin{aligned} p(y_{n+1} \mid x_{n+1}, x_{1:n}, y_{1:n}) &= \int p(y_{n+1} \mid x_{n+1}, \beta)\, p(\beta \mid x_{1:n}, y_{1:n})\, d\beta \\ &= \mathcal{N}(y_{n+1};\ \hat{\beta}_n x_{n+1},\ \sigma^2 + \sigma_n^2 x_{n+1}^2) \end{aligned}$$

The prediction is a Gaussian centered at $\hat{\beta}_n x_{n+1}$: the best estimate of β times the new input. The variance captures both measurement noise σ^2 and parameter uncertainty $\sigma_n^2 x_{n+1}^2$.

What a language model sees. If the context tokens encode $(x_1, y_1), \ldots, (x_n, y_n), x_{n+1}$, then the language model must predict y_{n+1}. It does not know β. It must infer β from the examples and use it to predict. The distribution it must learn to approximate, in order to have minimum cross-entropy loss, is exactly the Bayesian posterior predictive:

$$p(y_{n+1} \mid x_{n+1}, x_{1:n}, y_{1:n})$$

This is not a coincidence. Training on next-token prediction over diverse data forces the model to approximate the best possible predictor of the next token, which is the Bayesian predictor with the implicit parameters integrated out.



Numerical Illustration

Suppose $\sigma^2 = 1$, $\tau^2 = 1$, and the context provides three examples:

$$(x_1, y_1) = (1, 2.1), \quad (x_2, y_2) = (2, 3.9), \quad (x_3, y_3) = (3, 6.2)$$

These are consistent with $\beta \approx 2$. Compute:

$$\sum_i x_i^2 = 1 + 4 + 9 = 14, \quad \sum_i x_i y_i = 2.1 + 7.8 + 18.6 = 28.5$$

Posterior:

$$\frac{1}{\sigma_3^2} = 1 + 14 = 15, \quad \sigma_3^2 = 0.067$$

$$\hat{\beta}_3 = 0.067 \times 28.5 = 1.91$$

For a new query $x_4 = 4$:

$$p(y_4 \mid x_4 = 4, \text{context}) = \mathcal{N}(1.91 \times 4,\ 1 + 0.067 \times 16) = \mathcal{N}(7.63,\ 2.07)$$

After only three examples, the posterior has concentrated near the true slope $\beta = 2$, and the prediction 7.63 is close to the true value $y_4 \approx 8$. A language model performing in-context linear regression should output a distribution over tokens encoding y_4 that approximates this Gaussian — and empirically, large language models do exactly this.

11.4 Next Token Prediction as Bayesian Prediction

The General Argument

The regression example is a special case of a general principle. Consider any sequence $x_1, x_2, \ldots, x_T$ generated by a process with some latent structure z (a writing style, a topic, a speaker's knowledge, a programming language's syntax, a mathematical problem's solution):

$$z \sim p(z), \quad x_t \mid x_{<t}, z \sim p(x_t \mid x_{<t}, z)$$

The marginal distribution of the next token, integrating out the latent structure, is:

$$p(x_t \mid x_{<t}) = \int p(x_t \mid x_{<t}, z)\, p(z \mid x_{<t})\, dz$$

This is the **posterior predictive distribution**: the distribution of the next token averaged over the posterior distribution of the latent structure z given the context. The posterior $p(z \mid x_{<t})$ encodes everything the context tells us about the latent structure; the integral averages the prediction over all structures consistent with what has been observed.

A language model trained to minimize cross-entropy is trained to approximate $p(x_t \mid x_{<t})$ as closely as possible. In approximating this marginal, the model is — implicitly — doing Bayesian prediction: it is modeling the contribution of all possible latent structures, weighted by their posterior probability given the context.

Next token prediction is Bayesian prediction with implicit parameters.

$$p(x_t \mid x_{<t}) = \int p(x_t \mid x_{<t}, z)\, p(z \mid x_{<t})\, dz$$

The language model approximates the left-hand side. The right-hand side is a posterior predictive integral over latent structures z. A perfect language model is a

perfect Bayesian predictor, with the latent structure z integrated out. No explicit inference is required: the integration is implicit in the learned weights.

What are the Implicit Parameters?

The latent structure z can represent many different things depending on the context:

In in-context linear regression. $z = \beta$, the unknown slope. The context $(x_1, y_1), \ldots, (x_n, y_n)$ narrows the posterior over β, and the model predicts y_{n+1} by integrating over the remaining uncertainty.

In document generation. z might represent the topic, style, author identity, or subject matter. A few sentences from a legal document updates the posterior toward legal writing style; the model then predicts subsequent tokens consistent with that style.

In few-shot question answering. z might represent the rule or pattern exemplified by the provided demonstrations. Three examples of "French $\rightarrow$ English" translation update the posterior toward translation; the model then applies the inferred rule to the next French phrase.

In code generation. z might represent the programming language, the coding style, the algorithm being implemented. A few lines of Python with type annotations narrow the posterior toward a specific style; subsequent code is generated consistent with it.

In each case, the model has not been explicitly told the value of z. It infers z from the context by reading the tokens, and then generates subsequent tokens consistent with the inferred z. This is Bayesian inference from the context, followed by Bayesian prediction, implemented entirely in the forward pass of the transformer.

Why This Works: The Training Distribution Argument

Why would training on next-token prediction produce a model that behaves like a Bayesian predictor? The argument is information-theoretic.

The cross-entropy loss is:

$$\mathcal{L}(\theta) = -\mathbb{E}[\log p_\theta(x_t \mid x_{<t})]$$

This is minimized when $p_\theta(x_t \mid x_{<t}) = p(x_t \mid x_{<t})$ for all t and all contexts. The true conditional distribution $p(x_t \mid x_{<t})$ is the Bayesian posterior predictive. A model with sufficient capacity, trained on enough data, will therefore learn to approximate this posterior predictive — not because it was programmed to do Bayesian inference, but because that is the distribution that minimizes the training loss.

The model does not maintain an explicit posterior over z. Instead, the transformer's attention mechanism and feed-forward layers implement a distributed, implicit computation that achieves the same result: the output distribution $p_\theta(x_t \mid x_{<t})$ approximates the Bayesian posterior predictive.

Formally, if we write the true data-generating process as a mixture over latent structures:

$$p(x_t \mid x_{<t}) = \int p(x_t \mid x_{<t}, z)\, p(z \mid x_{<t})\, dz$$

then a language model that perfectly approximates $p(x_t \mid x_{<t})$ is a perfect approximation to this integral. The latent structures z are never explicitly represented; they are implicitly marginalized over by the learned weights.

In-Context Learning as Implicit Posterior Updating

This framework gives a precise interpretation of in-context learning. When a user provides examples $(x_1, y_1), \ldots, (x_n, y_n)$ in the prompt, they are providing evidence that updates the implicit posterior over the latent structure z:

$$p(z \mid x_{1:n}, y_{1:n}) \propto p(z) \prod_{i=1}^{n} p(y_i \mid x_i, z)$$

The model, in predicting the next token, is implicitly computing:

$$p(y_{n+1} \mid x_{n+1}, x_{1:n}, y_{1:n}) = \int p(y_{n+1} \mid x_{n+1}, z)\, p(z \mid x_{1:n}, y_{1:n})\, dz$$

This is Bayes rule, implemented implicitly by the transformer's forward pass. No weights are updated; no gradient is computed. The posterior update happens entirely through the attention mechanism reading the context.

In-context learning is implicit Bayesian inference. Providing examples in the prompt updates the implicit posterior over the latent task structure. The model predicts subsequent tokens by integrating over this posterior. The number of examples determines how concentrated the posterior is: more examples narrow the posterior and produce more task-specific predictions.

The role of scale. This Bayesian interpretation predicts that in-context learning should improve with model size: a larger model has more capacity to represent the implicit posterior over a richer space of latent structures. This is exactly what Brown et al. (2020) observed: in-context learning ability scales with model size in a way that qualitatively changes at certain thresholds. Small models show almost no in-context learning; large models show striking flexibility. The threshold corresponds, in the Bayesian interpretation, to the model having enough capacity to accurately approximate the posterior predictive over complex latent structures.

11.5 The Language Model as a Bayesian Reasoner

The Meta-Prior

Training on a large and diverse corpus is equivalent to learning a prior over all latent structures that could generate natural language. The training data — books, articles, code, conversations, scientific papers — is a sample from the distribution of human-generated text, which is in turn a sample from all possible latent structures (topics, styles, intentions, knowledge states) that human writers have.

By training on this corpus, the model learns a **meta-prior**: a distribution over the space of all possible tasks, rules, writing styles, and knowledge structures that could have generated any given text. When a user provides a context, the model updates this meta-prior to a posterior that is concentrated on structures consistent with the observed tokens.

This meta-prior is not explicitly constructed. It is learned implicitly from the statistics of the training data. A language model trained on English text has a meta-prior concentrated on English grammar, English vocabulary, and English semantic relationships. A model trained on code has a meta-prior concentrated on programming language syntax and algorithm patterns. A model trained on both has a richer meta-prior that can adapt to either context, depending on what the prompt reveals.

Limitations and Open Questions

The Bayesian interpretation of language models is intellectually satisfying and empirically supported, but it is not a complete theory. Several important limitations remain.

The approximation is imperfect. A real language model does not perfectly approximate the posterior predictive. It has finite capacity, was trained on finite data, and was optimized by an approximate algorithm (stochastic gradient descent). The Bayesian interpretation describes the ideal that the training objective aims for, not the model that is actually achieved.

The latent structure is not identified. The framework says the model integrates over latent structures z, but does not specify what z is. In the regression example, $z = \beta$ is explicit. For a general language model, z might be high-dimensional, compositional, and not reducible to any simple parameter. The "implicit parameters" are a useful conceptual device, not a concrete mathematical object.

Hallucination. When the model's implicit posterior is uncertain — when the context is ambiguous or the latent structure is poorly constrained — the model generates text that is coherent but factually wrong. This is not a failure of the Bayesian framework; a Bayesian predictor with high posterior uncertainty also produces wide, unreliable predictions. It is a failure of calibration: the model does not always know when it does not know.

The training distribution matters enormously. The meta-prior is determined by the training data. A model trained on biased, incorrect, or limited data has a biased

meta-prior and will make biased predictions. The prior's influence diminishes as context grows — more examples narrow the posterior — but it never vanishes entirely, and for short contexts or rare topics, the training distribution dominates.

The Bayesian picture of a large language model.

1. **Training** learns a meta-prior over latent structures from the statistics of the training corpus.
2. **The context** provides evidence that updates the meta-prior to a posterior over latent structures consistent with the observed tokens.
3. **Next token prediction** is posterior predictive inference: the next token is sampled from $p(x_t \mid x_{<t}) = \int p(x_t \mid x_{<t}, z)\, p(z \mid x_{<t})\, dz$.
4. **In-context learning** is implicit Bayesian updating: examples in the prompt concentrate the posterior on the relevant task structure without any weight update.
5. **Scale** enables this because a larger model can approximate richer posterior predictive distributions over more complex latent structures.

11.6 Connection to the Rest of This Book

The Bayesian interpretation of language models closes a circle that this book has been drawing from the beginning.

In Chapter 2, we introduced the physical and mental directions. The physical direction generates text from a latent structure: a writer with a topic, style, and intention produces tokens. The mental direction is inference: given the tokens, infer the latent structure. A language model runs the mental direction, inferring the implicit structure from the context.

In Chapter 5, we derived the posterior predictive distribution for parametric models: the Beta-Binomial predictive for the next coin flip, the Gaussian predictive for the next measurement. The language model's next-token distribution is the same object, generalized to an astronomical parameter space.

In Chapters 10 and 11, we developed MCMC and Langevin dynamics as methods for approximating posteriors over parameters. A language model does not run MCMC. Instead, it learns to approximate the marginal $p(x_t \mid x_{<t})$ directly, bypassing the need to explicitly maintain a posterior. This is variational inference in the limit of infinite capacity: instead of approximating the posterior $p(z \mid x_{<t})$ and integrating, the model directly approximates the integral.

In Chapter 12, we developed variational inference and the ELBO. The training objective of a language model — cross-entropy minimization — can be seen as maximizing a lower bound on the log marginal likelihood of the training data under the mixture

model $p(x) = \int p(x \mid z)p(z)\,dz$. The language model is a variational approximation to this mixture, with the variational distribution implicit in the weights rather than explicit in a parameterized family.

The language model is, in this sense, the endpoint of the computational hierarchy this book has developed. Exact Bayesian inference (conjugate models) is at the top, computationally trivial but applicable only to simple models. MCMC and variational inference approximate the posterior for complex models. The language model learns to approximate the posterior predictive for the most complex model of all: the distribution of human language.

Exercise 11.1. Consider the in-context regression setting with $y_i = \beta x_i + \varepsilon_i$, $\varepsilon_i \sim \mathcal{N}(0, 1)$, and prior $\beta \sim \mathcal{N}(0, 1)$.

1. Given context $(x_1, y_1) = (1, 3.1)$, $(x_2, y_2) = (2, 5.9)$, compute the posterior $p(\beta \mid x_{1:2}, y_{1:2})$ and the posterior predictive $p(y_3 \mid x_3 = 3, \text{context})$.

2. How does the posterior predictive change as $n \to \infty$ with $y_i/x_i \to 3$ for all i? What is the limiting distribution?

3. Explain why a language model trained on sequences of this form should approximate the posterior predictive rather than the MLE prediction $\hat{\beta}_{\text{MLE}} x_{n+1}$. When do the two predictions differ most?

Exercise 11.2. Let $x_1, x_2, \ldots$ be an exchangeable sequence generated by first drawing a latent parameter $z \sim p(z)$ and then drawing $x_t \mid z \overset{\text{iid}}{\sim} p(x \mid z)$ for each t.

1. Show that the optimal next-token predictor (minimizing cross-entropy) is the Bayesian posterior predictive $p(x_t \mid x_{<t}) = \int p(x \mid z)\, p(z \mid x_{<t})\, dz$.

2. Show that this posterior predictive converges to $p(x \mid z^*)$ as $t \to \infty$, where z^* is the true latent parameter. What does this say about the long-run behavior of a language model given an infinitely long context?

3. Explain what this implies about the role of the prior $p(z)$ (the meta-prior learned from training data) in short versus long contexts.

Exercise 11.3. The Bayesian interpretation predicts that in-context learning should improve with more examples up to a point, then plateau. Explain this in terms of the posterior $p(z \mid x_{<t})$:

1. Why does adding more examples initially improve predictions?

2. Why does the improvement eventually plateau?

3. What determines the plateau level, and how does the model's capacity affect it?

4. What happens when the in-context examples contradict the training distribution? How should the posterior $p(z \mid x_{<t})$ respond, and does in-context learning in practice behave this way?

Appendix: Three Classical Justifications for Bayesian Priors

The main text argues for Bayesian priors through decision theory and the ABC interpretation of search. This appendix presents the three classical arguments — Cox's theorem, Dutch book arguments, and Savage's axioms — in more detail. Each is mathematically substantive and historically important. Each also has genuine limitations that explain why the debate has not been settled by any one of them.

A.1 Cox's Theorem

The Goal

Richard Cox (1946) asked a foundational question: if an agent wants to reason about the plausibility of propositions — statements that may be true or false, but whose truth is not yet known — what mathematical structure must that reasoning have? He sought to derive probability theory from first principles, rather than postulate it.

The Setup

Let $p(A \mid B)$ denote the plausibility that proposition A is true, given that proposition B is known to be true. Cox imposed three requirements:

1. **Completeness.** Plausibility is represented by a single real number. If $p(A \mid B) > p(A' \mid B)$, then A is more plausible than A' given B.
2. **Consistency.** The plausibility of A given B should be determined by the plausibilities of the components. Specifically:
   - The plausibility of $\neg A$ given B is a function of $p(A \mid B)$ alone: $p(\neg A \mid B) = f(p(A \mid B))$.
   - The plausibility of $A \wedge B$ given C is a function of $p(B \mid C)$ and $p(A \mid B \wedge C)$ alone: $p(A \wedge B \mid C) = g(p(A \mid B \wedge C), p(B \mid C))$.
3. **Continuity.** The functions f and g are continuous and differentiable.

The Conclusion

Cox showed that under these requirements, f and g must satisfy functional equations whose only solutions (up to a monotone rescaling) are:

$$f(x) = 1 - x, \qquad g(x, y) = x \cdot y$$

That is, the negation rule and the product rule of probability theory are the unique consistent solutions. Since the sum rule follows from these two, probability theory is the only consistent system for reasoning about propositions under the given requirements.

Cox's theorem, informally. If you want to assign real numbers to the plausibility of propositions, and you want those assignments to be internally consistent and continuous, then your plausibilities must obey the rules of probability theory. The axioms of probability are not a choice; they are forced by the requirements of consistency.

Significance

Cox's theorem is remarkable because it derives probability from consistency requirements rather than from measure theory or betting. It suggests that probability is not merely a convention for describing random events, but the unique language for consistent reasoning under uncertainty.

Limitations

The desiderata are contestable. Why must plausibility be a single real number? Some philosophers argue that uncertainty should be represented by intervals or sets of probabilities. Why must $p(\neg A \mid B)$ depend only on $p(A \mid B)$? These are not self-evident requirements; they are design choices that lead to the desired conclusion. A committed non-Bayesian can deny one of them without logical contradiction.

The theorem says nothing about which prior to use. Even granting Cox's conclusion that plausibilities must satisfy the probability axioms, the theorem is silent about what probability to assign before any evidence is seen. It establishes the framework; it does not fill it. A scientist asking "what prior should I use for the speed of light?" gets no help from Cox's theorem.

Technical gaps. The original proof had gaps that required several decades to fully close. Jaynes (2003) and Van Horn (2003) provided more rigorous versions, but these require more careful statement of the conditions, and subtle counterexamples have been proposed at various points in the literature.

A.2 Dutch Book Arguments

The Goal

Frank Ramsey (1926) and Bruno de Finetti (1937) approached the foundations of probability from the direction of rational behavior rather than abstract consistency. Their key insight: if your beliefs are coherent, they can be represented as probabilities. If they are not, you can be exploited.

The Setup

An agent is asked to assign a fair betting price to every proposition A: the price $p(A)$ at which the agent is willing to either buy or sell a bet that pays \$1 if A is true and \$0 otherwise. The agent's willingness to bet at this price in either direction is called **coherence**: you cannot selectively choose to buy cheap bets and sell expensive ones.

The Dutch Book Theorem

Theorem 11.4 (Dutch Book Theorem)**.** *If an agent's betting prices $\{p(A)\}$ violate the axioms of probability, then there exists a collection of bets — a **Dutch book** — such that the agent will accept each bet individually but lose money with certainty, regardless of which propositions turn out to be true.*

For example, suppose an agent assigns $p(A) = 0.6$ and $p(\neg A) = 0.6$. Both prices are above 0.5, so the agent is willing to pay \$0.60 for a bet paying \$1 on A, and also \$0.60 for a bet paying \$1 on $\neg A$. But exactly one of A or $\neg A$ is true, so the agent collects \$1 but paid \$1.20: a guaranteed loss of \$0.20. The Dutch book exploits the incoherence $p(A) + p(\neg A) \neq 1$.

The converse (de Finetti's theorem) says that if an agent's prices are coherent — no Dutch book can be constructed — then they satisfy the probability axioms. Coherence is equivalent to being probabilistic.

Conditional Probability and Bayes Rule

The Dutch book argument extends to conditional probabilities. If an agent's conditional prices $p(A \mid B)$ are not related to their joint and marginal prices by the formula $p(A \mid B) = p(A \wedge B)/p(B)$, a Dutch book can be constructed using bets that are called off when B is false (**conditional bets**). Coherence therefore forces the conditional probability formula, and Bayes rule follows immediately.

Significance

The Dutch book argument is operationally concrete. It does not require abstract axioms about preferences or reasoning; it only requires that the agent not be exploitable.

And it connects the Bayesian framework directly to decision-making: your probabilities are your betting odds, and your updating rule is Bayes rule.

de Finetti's **representation theorem** extends the argument further. It shows that if an agent treats an infinite sequence of exchangeable events — events whose joint distribution is invariant to permutation — as if they were independently and identically distributed with some unknown parameter, then that parameter must have a prior distribution. Exchangeability implies the existence of a prior, not merely its permissibility.

Limitations

The betting operationalization is contestable. The framework identifies beliefs with willingness to bet. Many scientists are uncomfortable with this identification. Assigning a betting price to a physical constant — the speed of light, the mass of the Higgs boson — requires imagining a gamble on the outcome of a measurement, which conflates the unknown value of the constant with the randomness of the measurement process.

Dynamic Dutch books are more complex. The static Dutch book argument justifies the probability axioms at a single moment. Extending it to Bayesian updating over time (why should you update by Bayes rule when new evidence arrives?) requires additional assumptions and more complex argument. The diachronic Dutch book argument has been criticized on the grounds that it requires the agent to commit in advance to how they will update, which is a very strong requirement.

It prescribes coherence, not content. As with Cox's theorem, the Dutch book argument establishes that your prior must be a probability measure. It does not say what probability measure it should be.

A.3 Savage's Axioms

The Goal

Leonard Savage's 1954 book *The Foundations of Statistics* provided the most comprehensive axiomatic foundation for Bayesian statistics. Savage sought to derive both subjective probability and utility from primitive axioms about rational preferences among **acts**: functions from states of the world to consequences.

The Setup

The framework has three primitive concepts:

- A set $\mathcal{S}$ of **states of the world**, one of which is the true state. States are exhaustive and mutually exclusive.

- A set $\mathcal{C}$ of **consequences**: outcomes the agent cares about.
- A set $\mathcal{F}$ of **acts**: functions $f : \mathcal{S} \to \mathcal{C}$ mapping states to consequences.

The agent has a preference ordering $\succeq$ over acts: $f \succeq g$ means the agent weakly prefers act f to act g.

The Seven Axioms

Savage imposed seven axioms on $\succeq$:

1. **Weak order.** $\succeq$ is complete (any two acts can be compared) and transitive (if $f \succeq g$ and $g \succeq h$ then $f \succeq h$).
2. **Sure-thing principle.** If f and g agree on a set of states B^c, then the preference between f and g depends only on what they do on B. Formally: if $f \succeq g$ when restricted to B, the preference is unchanged by modifying both acts on B^c in the same way.
3. **State independence.** The preference between constant acts (acts that give the same consequence regardless of the state) does not depend on which event conditions the comparison.
4. **Non-degeneracy.** Not all acts are equally preferred; there exist $f \succ g$.
5. **Event ordering.** For any two events A and B, either A is "at least as probable" as B or vice versa, in a sense defined through preferences.
6. **Small event continuity.** For any act f and consequence c with $f \succ c$, there exists a partition of $\mathcal{S}$ fine enough that modifying f on any element of the partition to give consequence c still leaves the modified act preferred to c.
7. **Act-consequence independence.** If two acts give the same distribution over consequences, the agent is indifferent between them.

The Conclusion

Theorem 11.5 (Savage's Representation Theorem)**.** *If an agent's preferences satisfy the seven axioms, then there exists a unique probability measure P on $\mathcal{S}$ (the agent's subjective probability) and a utility function $U : \mathcal{C} \to \mathbb{R}$ (unique up to positive affine transformation) such that:*

$$f \succeq g \iff \mathbb{E}_P[U(f)] \geq \mathbb{E}_P[U(g)]$$

The agent behaves as if maximizing expected utility under the subjective probability P.

This is a powerful result. It derives both probability and utility from behavioral axioms about preferences, with no prior commitment to either concept. The prior P is not assumed; it is constructed from the agent's preference ordering.

Significance

Savage's theorem is the most complete axiomatic foundation for Bayesian statistics. It unifies probability and utility in a single framework, shows that rational preferences imply probabilistic beliefs, and provides a decision-theoretic justification for Bayesian updating: an agent who violates Bayes rule when updating beliefs upon new evidence will violate at least one of Savage's axioms.

The framework also connects naturally to the decision-theoretic arguments of Chapter 7. The Bayes estimator minimizes expected loss under the prior, which is exactly expected utility maximization in Savage's framework with $U = -L$.

Limitations

The sure-thing principle is empirically violated. The most criticized axiom is the sure-thing principle (Axiom 2). The **Allais paradox** (1953) provides a concrete example. Consider four acts:

Act	Win \$1M	Win \$5M	Win \$0
f_1	1.00	0.00	0.00
f_2	0.89	0.10	0.01
f_3	0.11	0.00	0.89
f_4	0.00	0.10	0.90

Most people prefer $f_1 \succ f_2$ (certainty of \$1M over a small chance of nothing) and $f_4 \succ f_3$ (better odds of \$5M). But the sure-thing principle requires these two preferences to be consistent in a way that can be shown to violate expected utility maximization for any utility function U and probability P. The Allais paradox is not a theoretical curiosity; it is a robust empirical finding replicated across many populations and contexts.

The framework describes idealized agents. Savage's axioms characterize perfectly rational behavior. Actual decision makers routinely violate them: they are loss-averse, they overweight small probabilities, they prefer certainty over gambles with higher expected value. Importing the framework as a normative justification for scientific inference requires accepting that scientists should reason as Savage's idealized agent reasons — a strong and contestable claim.

Infinite state spaces cause technical difficulties. Savage's original treatment assumed a finite number of consequences and required additional work to extend to infinite spaces. The extension requires topological assumptions that reintroduce the very measure-theoretic machinery the axioms were supposed to derive.

The prior is still not specified. As with Cox and Dutch books, Savage's theorem constructs the existence of a prior from the preference ordering, but it does not say what that prior should be. Two agents with different preferences will have different priors; the axioms do not adjudicate between them.

A.4 What the Three Arguments Establish Together

Despite their individual limitations, the three arguments converge on a coherent picture when taken together.

Cox's theorem establishes that *probability theory is the right language* for uncertainty: any consistent system of reasoning under uncertainty must obey the probability axioms.

The Dutch book argument establishes that *coherent beliefs are probabilistic*: an agent whose beliefs cannot be represented as probabilities is exploitable. De Finetti's exchangeability theorem additionally shows that *the prior exists whenever you believe in exchangeability*: if you treat observations as exchangeable, a prior over the unknown parameter is not optional but forced.

Savage's theorem establishes that *rational preferences imply probabilistic beliefs*: an agent who satisfies weak behavioral axioms behaves as if maximizing expected utility under a subjective prior.

Together they show that Bayesian reasoning is the coherent, rational, and consistent approach to inference under uncertainty. What they do not show — and what the ABC and decision-theoretic arguments of this book do show — is what the prior should be in any specific problem. The classical arguments justify the framework. Our arguments fill it.

The classical arguments and our arguments are complementary, not competing.

Argument	Establishes	Does not establish
Cox's theorem	Probability is the unique consistent language	Which prior to use
Dutch book	Coherent beliefs are probabilistic; exchangeability implies a prior	The content of the prior
Savage's axioms	Rational preferences imply a prior exists	Which prior reflects your preferences
Decision theory (Ch. 7)	A weight function is inevitable in any scalar comparison	*Prescribes* the prior from what you care about
ABC (Ch. 6)	The prior is a search strategy	*Prescribes* the prior from what you know

The classical arguments are deep and worth understanding. But a student who has worked through the ABC and decision-theoretic arguments of this book has some-

thing more: not just a proof that a prior must exist, but a constructive method for determining what it should be.

Further Reading

- **Cox (1946):** "Probability, frequency, and reasonable expectation," *American Journal of Physics* 14, 1–13. The original paper.
- **Jaynes (2003):** *Probability Theory: The Logic of Science*, Cambridge University Press. A comprehensive development of Cox's program, with many applications.
- **de Finetti (1937):** "Foresight: its logical laws, its subjective sources," translated in *Studies in Subjective Probability*, Wiley. The original Dutch book and exchangeability arguments.
- **Savage (1954):** *The Foundations of Statistics*, Wiley (2nd ed. Dover, 1972). The definitive axiomatic treatment.
- **Allais (1953):** "Le comportement de l'homme rationnel devant le risque," *Econometrica* 21, 503–546. The original paradox that challenges the sure-thing principle.
- **Berger (1985):** *Statistical Decision Theory and Bayesian Analysis*, Springer. The decision-theoretic perspective, including the complete class theorem.
- **Van Horn (2003):** "Constructing a logic of plausible inference," *International Journal of Approximate Reasoning* 33, 249–265. A rigorous modern version of Cox's theorem.

Appendix: Conjugate Priors

"The Boring Priors"

A prior is **conjugate** to a likelihood if the posterior belongs to the same parametric family as the prior. Conjugate priors are not necessarily the most realistic or most principled choice. They are, however, the computationally convenient choice: the posterior is available in closed form, with parameters updated by simple formulas. In Gibbs sampling, each conditional distribution is often a conjugate update of this kind. In variational inference, the mean field updates for conjugate models are analytic. And in teaching, conjugate models are the place where the algebra of Bayesian inference becomes completely transparent.

They are, in this sense, boring. They are also indispensable.

This appendix collects all the standard conjugate families in one place: the likelihood, the prior, the posterior, and the interpretation of each update. We group them by the type of data they describe.

B.1 Binary and Count Data

Beta-Binomial

Likelihood. $k \mid p, n \sim \text{Binomial}(n, p)$:

$$p(k \mid p) = \binom{n}{k} p^k (1-p)^{n-k}$$

Prior. $p \sim \text{Beta}(\alpha, \beta)$:

$$p(p) \propto p^{\alpha-1}(1-p)^{\beta-1}$$

Posterior.

$$p \mid k, n \;\sim\; \text{Beta}(\alpha + k,\; \beta + n - k)$$

Posterior mean.

$$\mathbb{E}[p \mid k] = \frac{\alpha + k}{\alpha + \beta + n}$$

Interpretation. The prior parameters (α, β) act as pseudo-counts: α prior successes and β prior failures. Each observed success adds 1 to α; each failure adds 1 to

β. The posterior mean is a weighted average of the prior mean $\alpha/(\alpha+\beta)$ and the data mean k/n. Special case: Beta$(1,1)$ = Uniform$[0,1]$ gives Laplace's rule of succession $\mathbb{E}[p \mid k] = (k+1)/(n+2)$.

Gamma-Poisson

Likelihood. $x \mid \lambda \sim \text{Poisson}(\lambda)$: the count of events in a fixed interval when events arrive at rate λ:

$$p(x \mid \lambda) = \frac{\lambda^x e^{-\lambda}}{x!}$$

Prior. $\lambda \sim \text{Gamma}(\alpha, \beta)$ with shape $\alpha > 0$ and rate $\beta > 0$:

$$p(\lambda) \propto \lambda^{\alpha-1} e^{-\beta\lambda}$$

Mean: α/β. Variance: α/β^2.

Posterior. After observing counts $x_1, \ldots, x_n$:

$$\lambda \mid x_1, \ldots, x_n \;\sim\; \text{Gamma}\left(\alpha + \sum_{i=1}^{n} x_i,\; \beta + n\right)$$

Posterior mean.

$$\mathbb{E}[\lambda \mid x] = \frac{\alpha + \sum x_i}{\beta + n}$$

Interpretation. The rate parameter β counts prior "exposure" (pseudo-observations) and α counts prior events. Each new observation adds 1 to the exposure and x_i to the event count.

Beta-Negative Binomial

Likelihood. $k \mid p$ is the number of failures before the r-th success:

$$p(k \mid p) = \binom{k+r-1}{k}(1-p)^k p^r$$

Prior. $p \sim \text{Beta}(\alpha, \beta)$.

Posterior. After observing k failures before r successes:

$$p \mid k \;\sim\; \text{Beta}(\alpha + r,\; \beta + k)$$

B.2 Continuous Data: Gaussian Models

Gaussian Mean (Known Variance)

Likelihood. $x_i \mid \mu \overset{\text{iid}}{\sim} \mathcal{N}(\mu, \sigma^2)$ with σ^2 known.

Prior. $\mu \sim \mathcal{N}(\mu_0, \tau^2)$.

Posterior.

$$\mu \mid x_1, \ldots, x_n \;\sim\; \mathcal{N}(\mu_n, \tau_n^2)$$

where:

$$\frac{1}{\tau_n^2} = \frac{1}{\tau^2} + \frac{n}{\sigma^2}, \qquad \mu_n = \tau_n^2\Big(\frac{\mu_0}{\tau^2} + \frac{n\bar{x}}{\sigma^2}\Big)$$

Interpretation. Posterior precision is the sum of prior precision and data precision. Posterior mean is the precision-weighted average of prior mean and sample mean. As $n \to \infty$, $\mu_n \to \bar{x}$: the data overwhelms the prior.

Gaussian Variance (Known Mean)

Likelihood. $x_i \mid \sigma^2 \overset{\text{iid}}{\sim} \mathcal{N}(\mu, \sigma^2)$ with μ known.

Prior. $\sigma^2 \sim$ Inverse-Gamma(α, β): the reciprocal of a Gamma random variable. Density:

$$p(\sigma^2) \propto (\sigma^2)^{-\alpha-1} \exp\!\left(-\frac{\beta}{\sigma^2}\right), \qquad \sigma^2 > 0$$

Mean: $\beta/(\alpha-1)$ for $\alpha > 1$. Variance: $\beta^2/((\alpha-1)^2(\alpha-2))$ for $\alpha > 2$.

Posterior.

$$\sigma^2 \mid x_1, \ldots, x_n \;\sim\; \text{Inverse-Gamma}\!\left(\alpha + \frac{n}{2},\; \beta + \frac{1}{2}\sum_{i=1}^{n}(x_i - \mu)^2\right)$$

Interpretation. The shape parameter α accumulates half the number of observations; the scale β accumulates half the sum of squared deviations. The prior parameters (α, β) act as 2α pseudo-observations with sum of squared deviations 2β.

Normal-Inverse-Gamma (Unknown Mean and Variance)

When both μ and σ^2 are unknown, the conjugate prior is the **Normal-Inverse-Gamma** distribution: $\sigma^2 \sim$ Inverse-Gamma(α, β) and $\mu \mid \sigma^2 \sim \mathcal{N}(\mu_0, \sigma^2/\kappa)$. The joint density is:

$$p(\mu, \sigma^2) \propto (\sigma^2)^{-\alpha-3/2} \exp\!\left(-\frac{2\beta + \kappa(\mu - \mu_0)^2}{2\sigma^2}\right)$$

Posterior. After observing $x_1, \ldots, x_n$ with sample mean $\bar{x}$ and sample variance $s^2 = \frac{1}{n}\sum(x_i - \bar{x})^2$:

$$\mu_n = \frac{\kappa\mu_0 + n\bar{x}}{\kappa + n}, \quad \kappa_n = \kappa + n$$

$$\alpha_n = \alpha + \frac{n}{2}, \quad \beta_n = \beta + \frac{ns^2}{2} + \frac{\kappa n(\bar{x} - \mu_0)^2}{2(\kappa + n)}$$

The posterior is Normal-Inverse-Gamma$(\mu_n, \kappa_n, \alpha_n, \beta_n)$.

Marginal posterior. Integrating out σ^2, the marginal posterior of μ is a Student-t distribution:

$$\mu \mid x_1, \ldots, x_n \;\sim\; t_{2\alpha_n}\left(\mu_n,\; \frac{\beta_n}{\alpha_n \kappa_n}\right)$$

with $2\alpha_n$ degrees of freedom. As $n \to \infty$, this approaches a Gaussian: the heavy tails of the t distribution reflect the additional uncertainty from not knowing σ^2.

B.3 Multivariate Gaussian Models

Multivariate Gaussian Mean (Known Covariance)

Likelihood. $x_i \mid \mu \overset{\text{iid}}{\sim} \mathcal{N}_d(\mu, \Sigma)$ with Σ known.

Prior. $\mu \sim \mathcal{N}_d(\mu_0, \Lambda^{-1})$ where $\Lambda = \tau^{-2} I$ is the prior precision matrix.

Posterior.

$$\mu \mid x_1, \ldots, x_n \;\sim\; \mathcal{N}_d(\mu_n, \Lambda_n^{-1})$$

where:

$$\Lambda_n = \Lambda + n\Sigma^{-1}, \qquad \mu_n = \Lambda_n^{-1}(\Lambda \mu_0 + n\Sigma^{-1}\bar{x})$$

The posterior precision is the sum of the prior precision and the data precision, exactly as in the scalar case, but now in matrix form.

Inverse-Wishart Distribution

The **Inverse-Wishart** distribution is the multivariate generalization of the Inverse-Gamma, and is the conjugate prior for a Gaussian covariance matrix.

A $d \times d$ positive definite matrix Σ has the Inverse-Wishart distribution $\Sigma \sim \mathcal{W}^{-1}(\Psi, \nu)$ with scale matrix Ψ (positive definite, $d \times d$) and degrees of freedom $\nu > d-1$ if:

$$p(\Sigma) \propto |\Sigma|^{-(\nu+d+1)/2} \exp\left(-\frac{1}{2}\text{tr}(\Psi\Sigma^{-1})\right)$$

Mean: $\mathbb{E}[\Sigma] = \Psi/(\nu - d - 1)$ for $\nu > d + 1$.

Intuition. Think of Ψ as a prior sum of squared deviations (a $d \times d$ matrix version of β in the Inverse-Gamma) and ν as the number of prior pseudo-observations.

Multivariate Gaussian Covariance (Known Mean)

Likelihood. $x_i \mid \Sigma \overset{\text{iid}}{\sim} \mathcal{N}_d(0, \Sigma)$.

Prior. $\Sigma \sim \mathcal{W}^{-1}(\Psi, \nu)$.

Posterior.

$$\Sigma \mid x_1, \ldots, x_n \;\sim\; \mathcal{W}^{-1}(\Psi + S,\; \nu + n)$$

where $S = \sum_{i=1}^{n} x_i x_i^\top$ is the sample scatter matrix.

Interpretation. Each observation contributes its outer product $x_i x_i^\top$ to the scale matrix, augmenting the prior scatter Ψ. The degrees of freedom ν increase by one per observation, just as the shape parameter of the Inverse-Gamma increases by $1/2$ per scalar observation.

Normal-Inverse-Wishart (Unknown Mean and Covariance)

The joint conjugate prior for (μ, Σ) in a multivariate Gaussian model is the **Normal-Inverse-Wishart**:

$$\Sigma \sim \mathcal{W}^{-1}(\Psi, \nu), \qquad \mu \mid \Sigma \sim \mathcal{N}_d\left(\mu_0,\ \frac{\Sigma}{\kappa}\right)$$

Posterior. After observing $x_1, \ldots, x_n$:

$$\kappa_n = \kappa + n, \quad \mu_n = \frac{\kappa \mu_0 + n\bar{x}}{\kappa + n}, \quad \nu_n = \nu + n$$

$$\Psi_n = \Psi + S + \frac{\kappa n}{\kappa + n}(\bar{x} - \mu_0)(\bar{x} - \mu_0)^\top$$

where $S = \sum_i (x_i - \bar{x})(x_i - \bar{x})^\top$.

The third term in Ψ_n is the contribution from the discrepancy between the sample mean $\bar{x}$ and the prior mean μ_0: if the data mean is far from the prior mean, this increases the posterior scale matrix, reflecting additional uncertainty.

Wishart Distribution

The **Wishart** distribution $W \sim \mathcal{W}(\Sigma, \nu)$ is the conjugate prior for a Gaussian *precision* matrix $\Omega = \Sigma^{-1}$. Density:

$$p(W) \propto |W|^{(\nu - d - 1)/2} \exp\left(-\frac{1}{2}\mathrm{tr}(\Sigma^{-1} W)\right)$$

Mean: $\mathbb{E}[W] = \nu\Sigma$. If $W \sim \mathcal{W}(\Sigma, \nu)$ then $W^{-1} \sim \mathcal{W}^{-1}(\Sigma^{-1}, \nu)$.

B.4 Categorical and Multinomial Data

Dirichlet-Multinomial

The **Dirichlet** distribution is the multivariate generalization of the Beta and is conjugate to the Multinomial likelihood.

Likelihood. Observe n draws from a categorical distribution over K categories, with category k having probability π_k. The counts $n_1, \ldots, n_K$ with $\sum_k n_k = n$ follow:

$$p(n_1, \ldots, n_K \mid \pi) = \binom{n}{n_1, \ldots, n_K} \prod_{k=1}^{K} \pi_k^{n_k}$$

Prior. $\pi = (\pi_1, \ldots, \pi_K) \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_K)$:

$$p(\pi) \propto \prod_{k=1}^{K} \pi_k^{\alpha_k - 1}, \qquad \pi_k \geq 0,\ \sum_k \pi_k = 1$$

Mean: $\mathbb{E}[\pi_k] = \alpha_k / \sum_j \alpha_j$. The symmetric Dirichlet $\alpha_1 = \cdots = \alpha_K = \alpha$ has mean $1/K$ for each category and concentrates near the uniform distribution for large α.

Posterior.

$$\pi \mid n_1, \ldots, n_K \ \sim\ \text{Dirichlet}(\alpha_1 + n_1,\ \ldots,\ \alpha_K + n_K)$$

Interpretation. Identical to the Beta-Binomial in every respect, generalized to K categories. The prior parameters α_k are pseudo-counts. Each observed count n_k adds to the corresponding pseudo-count. The posterior mean is $(\alpha_k + n_k)/(\sum_j \alpha_j + n)$: a weighted average of the prior mean and the empirical frequency.

In the Gibbs sampler for mixture models. With K components and assignment counts $n_1, \ldots, n_K$, the mixing weights update as:

$$\pi \mid z \ \sim\ \text{Dirichlet}(\alpha_1 + n_1, \ldots, \alpha_K + n_K)$$

This is exactly the conjugate update, applied to the current assignment counts.

B.5 The Dirichlet Process

The conjugate models above all share a common limitation: the number of components, categories, or clusters must be fixed in advance. In practice, we often do not know how many components a mixture model should have, or how many distinct values a discrete distribution should assign positive probability to. The **Dirichlet process** is a nonparametric generalization of the Dirichlet distribution that places a prior directly over the space of discrete probability measures, allowing the number of components to grow with the data.

From Dirichlet to Dirichlet Process

Recall the finite Dirichlet. A draw $\pi \sim \text{Dirichlet}(\alpha/K, \ldots, \alpha/K)$ from a symmetric K-dimensional Dirichlet (with each parameter equal to α/K) is a probability vector

over K categories. As $K \to \infty$ with α fixed, the distribution over probability vectors converges to the **Dirichlet process**.

Formally, let H be a probability distribution on some space Θ (the **base measure**: where the atoms will be placed) and $\alpha > 0$ be a **concentration parameter**. A random probability measure G on Θ follows the Dirichlet process $\mathrm{DP}(\alpha, H)$ if, for every finite partition $\{A_1, \ldots, A_K\}$ of Θ:

$$(G(A_1), \ldots, G(A_K)) \sim \mathrm{Dirichlet}(\alpha H(A_1), \ldots, \alpha H(A_K))$$

The Stick-Breaking Construction

The Dirichlet process has a beautiful constructive representation known as **stick-breaking** (Sethuraman, 1994). Draw:

$$v_k \overset{\text{iid}}{\sim} \mathrm{Beta}(1, \alpha), \quad k = 1, 2, 3, \ldots$$

Define the weights:

$$\pi_k = v_k \prod_{j=1}^{k-1} (1 - v_j)$$

This is the stick-breaking metaphor: start with a stick of length 1. Break off a fraction v_1 (giving weight $\pi_1 = v_1$). From the remainder, break off fraction v_2 (giving weight $\pi_2 = v_2(1 - v_1)$). Continue indefinitely. The weights $\pi_1, \pi_2, \ldots$ sum to 1 almost surely.

Now draw atom locations $\theta_k \overset{\text{iid}}{\sim} H$ independently of the v_k. The random measure:

$$G = \sum_{k=1}^{\infty} \pi_k \, \delta_{\theta_k}$$

is a draw from $\mathrm{DP}(\alpha, H)$. It is almost surely a discrete distribution, regardless of whether H is continuous. This is the key feature: the Dirichlet process always produces a discrete distribution with countably many atoms, even when the base measure H is continuous.

The Role of α

The concentration parameter α controls how spread the weights are across atoms:

- **Small α:** the first few breaks take most of the stick. The measure G concentrates on a small number of atoms. In a mixture model, this favors solutions with few clusters.
- **Large α:** each break takes a small fraction. The weights are spread across many atoms. In a mixture model, this favors many small clusters.

- **The expected number of distinct components** in n observations is approximately $\alpha \log(1+n/\alpha)$: logarithmic growth in n. More data tends to reveal more clusters, but the growth is slow.

The Chinese Restaurant Process

The **Chinese restaurant process** (CRP) is the marginal distribution over cluster assignments implied by the Dirichlet process, obtained by integrating out G. Imagine n customers entering a restaurant with infinitely many tables. Customer 1 sits at table 1. Customer i (for $i \geq 2$):

- Sits at an occupied table k with probability $n_k/(i-1+\alpha)$, where n_k is the current number of customers at table k.
- Sits at a new table with probability $\alpha/(i-1+\alpha)$.

The resulting partition of n customers into tables (clusters) is the CRP. Popular tables attract more customers (**rich get richer**), but new tables are always possible. The CRP is exchangeable: the probability of any given partition does not depend on the order in which customers arrive.

Dirichlet Process Mixture Models

In a **Dirichlet process mixture model**, each observation x_i is generated by:

$$G \sim \mathrm{DP}(\alpha, H), \quad \theta_i \mid G \sim G, \quad x_i \mid \theta_i \sim F(\theta_i)$$

where $F(\theta)$ is a parametric family (e.g., $\mathcal{N}(\mu, \sigma^2)$). Because G is discrete, multiple observations share the same θ: this creates clustering automatically, without fixing the number of components. The number of clusters grows logarithmically with n, adapting to the data.

Inference is typically performed by collapsed Gibbs sampling using the CRP representation. The assignment of observation i to a cluster follows:

$$p(z_i = k \mid z_{-i}, x_i, \ldots) \propto \begin{cases} n_{-i,k}\, p(x_i \mid \theta_k) & \text{existing cluster } k \\ \alpha \int p(x_i \mid \theta)\, H(\theta)\, d\theta & \text{new cluster} \end{cases}$$

where $n_{-i,k}$ is the number of observations other than i currently in cluster k. The integral for a new cluster is the marginal likelihood of x_i under the base measure H, which is analytic for conjugate H.

The Dirichlet process in one sentence. The Dirichlet process is the infinite-dimensional limit of the Dirichlet distribution: a prior over discrete probability measures that allows the number of components to be inferred from the data rather than fixed in advance, with the concentration parameter α controlling the expected number of clusters and the base measure H specifying where cluster parameters

come from.

B.6 Summary Table

Likelihood	Prior	Posterior	Updated Parameters
Binomial(n, p)	Beta(α, β)	Beta	$(\alpha{+}k,\ \beta{+}n{-}k)$
Poisson(λ)	Gamma(α, β)	Gamma	$(\alpha{+}\sum x_i,\ \beta{+}n)$
$\mathcal{N}(\mu, \sigma^2)$, σ^2 known	$\mathcal{N}(\mu_0, \tau^2)$	$\mathcal{N}$	Precision-weighted avg
$\mathcal{N}(\mu, \sigma^2)$, μ known	Inv-Gamma(α, β)	Inv-Gamma	$(\alpha{+}n/2,\ \beta{+}\frac{1}{2}\sum(x_i{-}\mu)^2)$
$\mathcal{N}(\mu, \sigma^2)$, both unknown	Normal-Inv-Gamma	Normal-Inv-Gamma	Four-parameter update
$\mathcal{N}_d(\mu, \Sigma)$, Σ known	$\mathcal{N}_d(\mu_0, \Lambda^{-1})$	$\mathcal{N}_d$	Matrix precision sum
$\mathcal{N}_d(\mu, \Sigma)$, μ known	Inv-Wishart(Ψ, ν)	Inv-Wishart	$(\Psi{+}S,\ \nu{+}n)$
$\mathcal{N}_d(\mu, \Sigma)$, both unknown	Normal-Inv-Wishart	Normal-Inv-Wishart	Four-parameter update
Multinomial(π)	Dirichlet(α)	Dirichlet	$(\alpha_k{+}n_k)_{k=1}^K$
Categorical (infinite)	Dirichlet Process	Dirichlet Process	CRP update

The table is organized from simplest to most complex. Every row is the same operation: the prior pseudo-counts are augmented by the sufficient statistics of the data. In the scalar cases, the sufficient statistic is the sample mean or sum. In the matrix cases, it is the scatter matrix S. In the Dirichlet case, it is the vector of counts. In the Dirichlet process, it is the table assignment in the Chinese restaurant. The algebra changes; the logic does not.