Ch 3 Markov Chain Basics

In this chapter, we introduce the background of MCMC computing

Topics:

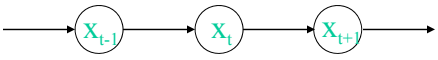
1. What is a Markov chain?
2. Some examples for simulation, approximate counting, Monte Carlo integration, optimization.
3. Basic concepts in MC design: transition matrix, positive recurrence, ergodocity.

Reading materials: Bremaud Ch 2.1-2.4, Ch 3.3-3.4.

What is Markov Chain?

A **Markov chain** is a mathematical model for stochastic systems whose states, discrete or continuous, are governed by a transition probability. The current state in a Markov chain only depends on the most recent previous states, e.g. for a 1st order Markov chain.

$$x_t|x_{t-1},...,x_0 \sim P(x_t|x_{t-1},...,x_0) = P(x_t|x_{t-1})$$



The **Markovian property** means "locality" in space or time, such as Markov random fields and Markov chain. Indeed, a discrete time Markov chain can be viewed as a special case of the Markov random fields (causal and 1-dimensional).

A **Markov chain** is often denoted by (Ω, ν, K) for state space, initial and transition prob.

What is Monte Carlo ?

Monte Carlo is a small hillside town in Monaco (near Italy) with casino since 1865 like Los Vegas in the US. It was picked by a physicist Fermi (Italian born American) who was among the first using the sampling techniques in his effort building the first man-made nuclear reactors in 1942.

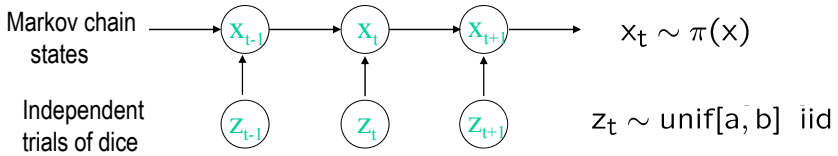
What is in common between a **Markov chain** and the **Monte Carlo casino**?

They are both driven by random variables --- using dice.

Monte Carlo casino

What is Markov Chain Monte Carlo ?

MCMC is a **general purpose technique** for generating **fair samples** from a probability in high-dimensional space, using random numbers (dice) drawn from uniform probability in certain range. A Markov chain is designed to have $\pi(x)$ being its **stationary (or invariant) probability.**



This is a non-trivial task when $\pi(x)$ is very complicated in very high dimensional spaces !

What is Sequential Monte Carlo ?

Discuss the difference between MCMC and SMC here.

Common: represent a probability distribution by a set of examples with weights (equal or not).

Discussion: how is this related to search?

MCMC as a general purpose computing technique

Task 1: Simulation: draw fair (typical) samples from a probability which governs a system.

$$x \sim \pi(x), \text{ s is a configuration.}$$

Task 2: Integration / computing in very high dimensions, i.e. to compute

$$c = E[f(x)] = \int \pi(x) f(x) ds$$

Task 3: Optimization with an annealing scheme

$$x^* = \text{argmax } \pi(x)$$

Task 4: Learning:

unsupervised learning with hidden variables (simulated from posterior) or MLE learning of parameters $p(x; \theta)$ needs simulations as well.

Task 1: Sampling and simulation

For many systems, their states are governed by some probability models. e.g. in statistical physics, the microscopic states of a system follows a Gibbs model given the macroscopic constraints. The fair samples generated by MCMC will show us what states are *typical* of the underlying system. In computer vision, this is often called "*synthesis*" ---the visual appearance of the simulated images, textures, and shapes, and it is a way to *verify* the sufficiency of the underlying model.

Suppose a system state x follows some global constraints.

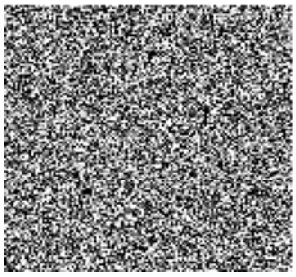
$$x \in \Omega = \{x : H_i(x) = h_i, i = 1, 2, ..., K\}$$

Hi(s) can be a hard (logic) constraints (e.g. the 8-queen problem), macroscopic properties (e.g. a physical gas system with fixed volume and energy), or statistical observations (e.g the Julesz ensemble for texture).

Ex. 1 Simulating noise image

We define a "noise" pattern as a set of images with fixed mean and variance.

$$\text{noise} = \Omega(\mu, \sigma^2) = \{I_\Lambda : \lim_{\Lambda \to Z^2} \frac{1}{|\Lambda|} \sum_{(i,j)\in\Lambda} I(i,j) = \mu, \lim_{\Lambda \to Z^2} \frac{1}{|\Lambda|} \sum_{(i,j)\in\Lambda} (I(i,j) - \mu)^2 = \sigma^2 \}$$

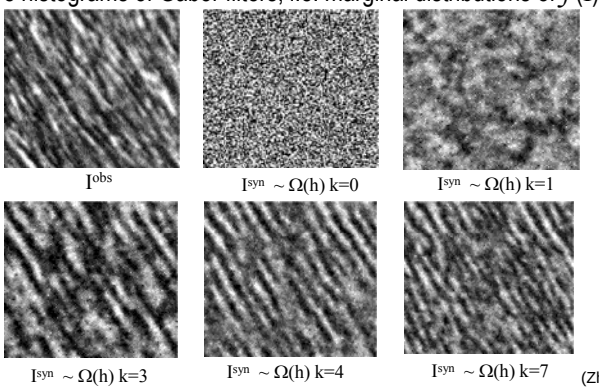


This image example is a "typical image" of the Gaussian model.

Ex. 2 Simulating typical textures by MCMC

$$\text{a texture} = \Omega(h_c) = \{ I : \lim_{\Lambda \to Z^2} \frac{1}{|\Lambda|} \sum_{(i,j) \in \Lambda} h(I_{(i,j)}) = h_c , \ | h_c | = k \}$$

H_c are histograms of Gabor filters, i.e. marginal distributions of $f(I)$



I^{obs} $I^{syn} \sim \Omega(h)$ k=0 $I^{syn} \sim \Omega(h)$ k=1

$I^{syn} \sim \Omega(h)$ k=3 $I^{syn} \sim \Omega(h)$ k=4 $I^{syn} \sim \Omega(h)$ k=7

(Zhu et al, 1996-01)

Task 2: Scientific computing

In scientific computing, one often needs to compute the integral in very high dimensional space.

Monte Carlo integration,

e.g.

1. estimating the expectation by empirical mean.
2. importance sampling

Approximate counting (so far, not used in computer vision)

e.g.

1. how many non-self-intersecting paths are in a 2 n x n lattice of length N?
2. estimate the value of π by generating uniform samples in a unit square.

Ex 3: Monte Carlo integration

Often we need to estimate an integral in a very high dimensional space Ω,

$$c = \int_{\Omega} \pi(x) f(x) dx$$

We draw N samples from $\pi(x)$,

$$x_1, x_2, ..., x_N \sim \pi(x)$$

Then we estimate C by the sample mean

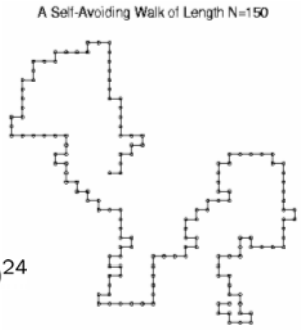
$$\hat{c} = \frac{1}{N} \sum_{i=1}^{N} f(x_i)$$

For example, we estimate some statistics for a Julesz ensemble $\pi(x;\theta)$,

$$C(\theta) = \int_{\Omega} \pi(x;\theta) H(x) dx$$

Ex 4: Approximate counting in polymer study

For example, what is the number K of Self-Avoiding-Walks in an n x n lattice?



Denote the set of SAWs by $\Omega_{n^2} = \{r\}$

An example of n=10. (Persi Diaconis)

The estimated number by Knuth was $(1.6 \pm 0.3) \times 10^{24}$

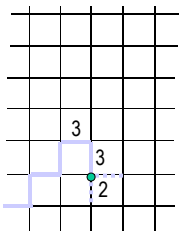
The truth number is 1.56875×10^{24}

(Note that there are a variety of different definitions of SAWs: Start from the lower-left corner, the ending could be of (i) any lengths, (ii) fixed length n, or (iii) ending at the upper-right corner. The number above is for case (iii).

Ex 4: Approximate counting in polymer study

Computing K by MCMC simulation

$$K = \sum_{r\in\Omega_{n2}} 1 = \sum_{r\in\Omega_{n2}} \frac{1}{p(r)} p(r)$$
$$= E[\frac{1}{p(r)}]$$
$$\approx \frac{1}{M}\sum_{i=1}^{M} \frac{1}{p(r_i)}$$



Sampling SAWs r_i by random walks (roll over when it fails).

$$p(r) = \prod_{j=1}^{m} \frac{1}{k(j)}$$

Task 3: Optimization and Bayesian inference

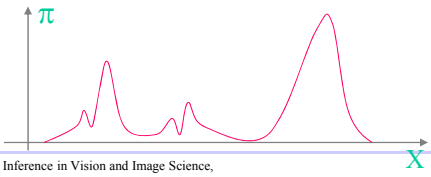
A basic assumption, since Helmholtz (1860), is that biologic and machine vision compute the most probable interpretation(s) from input images.

Let I be an image and X be a semantic representation of the world.

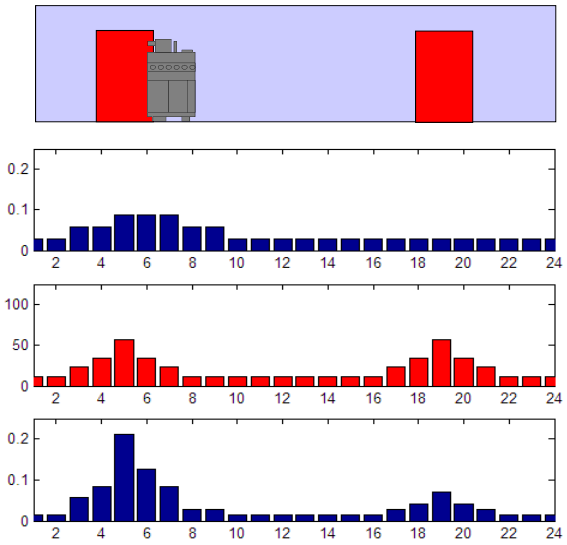
$$X^* = \arg\max \pi(X|I)$$

In statistics, we need to sample from the posterior and keep multiple solutions.

$$(X_1, X_2, ..., X_k) \sim \pi(X|I)$$



Example 5: Robot Localization



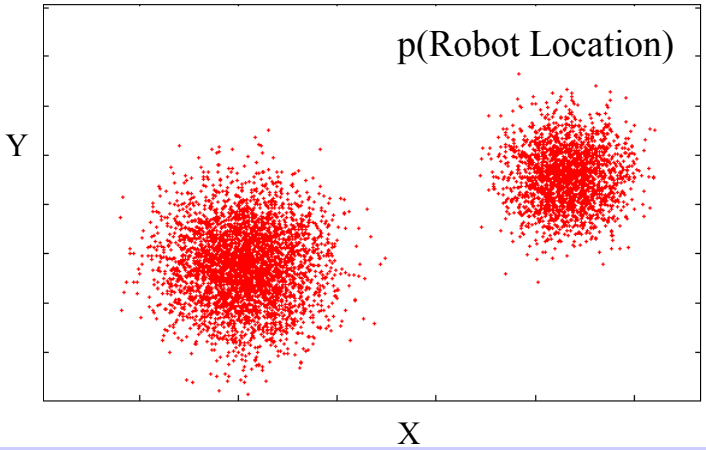
Prior P(x)

Likelihood L(x;z)

Posterior P(x|z)

Example 5: Robot Localization

Sampling as Representation

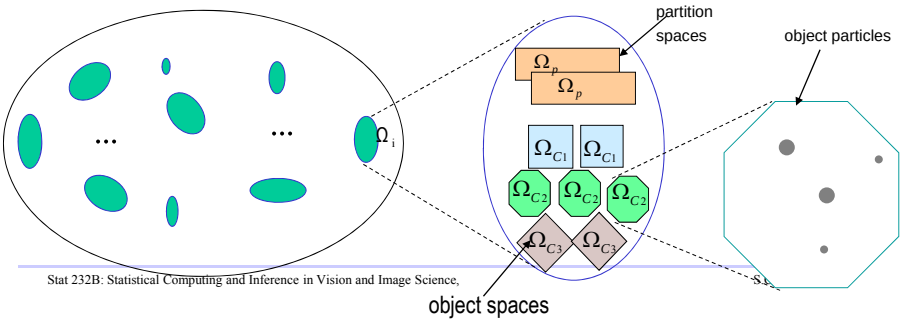


Traversing Complex State Spaces

1. The state space Ω in computer vision often has a large number of sub-spaces of varying dimensions and structures, because of the diverse visual patterns in images.

2. Each sub-space is a product of
   some *partition (coloring) spaces* ---- what go with what?
   some *object spaces* ---- what are what?

3. The posterior has low entropy, the *effective volume* of the search space is relatively small !



Summary

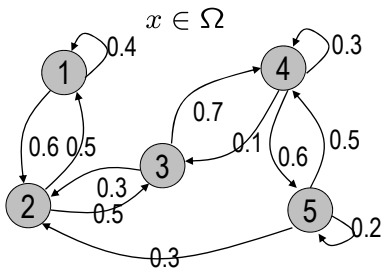
1. MCMC is a general purpose technique for sampling from complex probabilistic models.

2. In high dimensional space, sampling is a key step for
   (a) modeling (simulation, synthesis, verification)
   (b) learning (estimating parameters)
   (c) estimation (Monte Carlo integration, importance sampling)
   (d) optimization (together with simulated annealing).

2. As Bayesian inference have become a major framework in computer vision, the MCMC technique is a useful tool of increasing importance for more and more advanced vision models.

A Toy Example

Suppose there are 5 families in an island. Suppose there is 1,000,000 token as their currency, and we normalize them to 1. Let the state x be the wealth over the years. Each family will trade with some other families for goods. For example, family 1 will spend 60% of their income to buy from family 2, and save 40% income, and so on. The question is: how will the fortune be distributed among the families after a number of years? To put the question in the other way, suppose we mark one token In a special color (say, red). After a number of years, who will own this token?



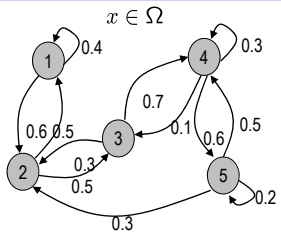
A Markov chain formulation

$$(\Omega,\ K \text{ or } P,\ \nu_o)$$

1. State space 2. Transition kernel. 3. Initial probability.

$$K = \begin{pmatrix} 0.4 & 0.6 & 0.0 & 0.0 & 0.0 \\ 0.5 & 0.0 & 0.5 & 0.0 & 0.0 \\ 0.0 & 0.3 & 0.0 & 0.7 & 0.0 \\ 0.0 & 0.0 & 0.1 & 0.3 & 0.6 \\ 0.0 & 0.3 & 0.0 & 0.5 & 0.2 \end{pmatrix}$$

Target Distribution



$$\lim_{n\to\infty} p_o K^n \to \pi$$

year										
1	1.0	0.0	0.0	0.0	0.0		0.0	0.0	1.0	0.0	0.0
2	0.4	0.6	0.0	0.0	0.0		0.0	0.3	0.0	0.7	0.0
3	0.46	0.24	0.30	0.0	0.0		0.15	0.0	0.22	0.21	0.42
4	...						...				
5											
6	0.23	0.21	0.16	0.21	0.17		0.17	0.16	0.16	0.26	0.25
	0.17	0.20	0.13	0.28	0.21		0.17	0.20	0.13	0.28	0.21

Invariant probabilities

Under certain conditions for the finite state Markov chains, the Markov chain state converges to an invariant probability

$$\lim_{n\to\infty} \mu_o P^n \to \mu$$

In Bayesian inference, we are given a target probability μ, our objective is to design a Markov chain kernel P so that P has a unique invariant probability μ.

There are infinity number of P's that have the same invariant probability.

Questions

1. What are the conditions for P?
   (stochastic, irreducible, aperiodic, global/detailed balance, ergodicity and positive recurrence, …)

2. How do we measure the effectiveness (i.e. convergence) ?
   (first hitting time, mixing time)

3. How do we diagnose convergence?
   (exact sampling techniques for some special chains)

Choice of K

Markov Chain Design:

(1) K is an irreducible (egordic) stochastic matrix (each row sum to 1).

(2) K is aperiodic (with only one eigen-value to be 1).

(3) Detailed balance $p(i)K_{ij} = p(j)K_{ji}$

There are almost infinite number of ways to construct K given a $\boldsymbol{\pi}$.

2N equations with N x N unknowns (global balance), or

$N^2/2+N$ equations with r x r unknowns (detailed balance)

Different Ks have different performances.

Communication Class

A state j is said to be *accessible* from state i if there exists M such $K_{ij}(M)>0$

$$i \longrightarrow j \qquad K_{ij}(M) = \sum_{i_1,\ldots,i_{M-1}} K_{ii_1}\ldots K_{i_{M-1}j} \;\; K_{ij}(M) > 0$$

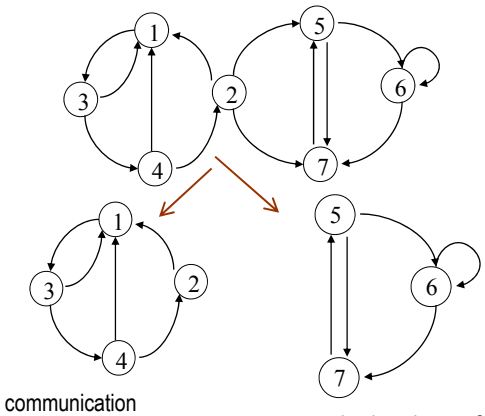
$$i \longleftrightarrow j \qquad i \text{ and } j \text{ are accessible to each other}$$

Communication relation $\longleftrightarrow$ generates a partition of the sate space into disjoint equivalence classes called *communication classes*.

Definition:
A Markov chain is *irreducible* if its matrix K has only one communication class.

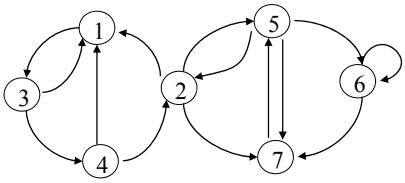
Irreducibility

If there exists only one communication class then we call its transition graph to be irreducible (**ergodic**).



communication classes 1

communication classes 2

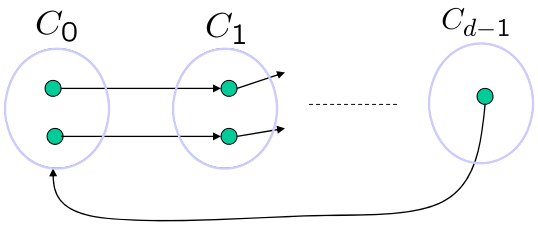


Irreducible MC

Periodic Markov Chain

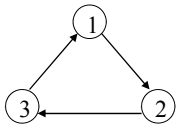
For any irreducible Markov chain, one can find a unique partition of graph G into d classes:

$$C_0, \ldots, C_{d-1}, \quad i \in C_k \qquad \sum_{j \in C_k} K_{ij} = 1$$



Periodic Markov Chain

An example:



$$K = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

The Markov Chain has period 3 and it alternates at three distributions:

$$(1\ 0\ 0) \rightarrow (0\ 1\ 0) \rightarrow (0\ 0\ 1)$$

An irreducible stochastic matrix K has period d, then
K has one communication class, but K^d has d communication classes.

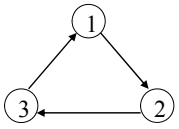
Stationary Distribution

$$\pi = \pi K$$

There maybe many stationary distributions w.r.t K.

Even there is a stationary distribution, Markov chain may not always converge to it.

$$\pi = (\tfrac{1}{3}\ \tfrac{1}{3}\ \tfrac{1}{3}) \qquad K = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \qquad (\tfrac{1}{3}\ \tfrac{1}{3}\ \tfrac{1}{3}) \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} = (\tfrac{1}{3}\ \tfrac{1}{3}\ \tfrac{1}{3})$$



$$(1\ 0\ 0) \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} = (0\ 1\ 0)$$

Markov Chain Design

Given a target distribution π, we want to design an irreducible and aperiodic K

$$\pi K = \pi \qquad \text{and } K \text{ has small } \lambda_{SLEM}$$

The easiest would be: $K = \begin{pmatrix} \pi \\ \vdots \\ \pi \end{pmatrix}$ then any $pK = \pi$ $\lambda_{SLEM}(K) = 0$

But in general *x* is in a big space and we don't know the landscape of π, though we can compute each π(x).

Sufficient Conditions for Convergence

Irreducible (ergodic):

$$\forall i \leftrightarrow j, K_{ij}(M) > 0 \ \ and \ \ K_{ji}(M) > 0$$

Detailed Balance: $\pi(i)K_{ij} = \pi(j)K_{ji}$

Detailed balance implies stationarity:

$$\pi K = \sum_i \pi(i)K_i = \sum_i \pi(i)(K_{i1}, ..., K_{in})$$

$$= \sum_i (\pi(j)K_{1i}, ..., \pi(n)K_{ni}) = \pi$$

The Perron-Frobenius Theorem

For any primitive (irreducibility + aperiodicity) r x r stochastic matrix P, P has eigen-values

$$1 = \lambda_1 > |\lambda_2| > \cdots > |\lambda_r|$$

Each eigen-value has left and right eigen-vectors (μ_i, ν_i)
With $\nu_1 = 1, \mu_1 = \pi$

Then $$P^n = 1 \cdot \pi' + O(n^{m_2 - 1}|\lambda_2|^n)$$

Where m2 is the algebraic multiplicity of λ_2, i.e. m_2 eigen-values that have the same modulus.

Then obviously, the convergence rate is $\lambda_{\text{slem}} = |\lambda_2|$

The Perron-Frobenius Theorem

Now, why do we need irreducibility and aperiodicity?

1, If P is not irreducible, and has C communication classes.
then the first eigen value 1 has C algebraic and geometric multiplicities (eigen-vectors)
Thus it does not have a unique invariant probability.

2, If P is irreducible but has period d >1, then there are d distinct eigen values
with modius 1, namely, the d-th roots of unity.

Convergence measures

The first hitting time of a state i by a Markov chain MC is

$$\tau_{hit}(i) = \inf\{ n \geq 1; x_n = i, x_0 \sim \mu_0\}$$

The first return time of a state i by a Markov chain MC is

$$\tau_{ret}(i) = \inf\{ n \geq 1;\ x_n = i,\ x_0 = i\}$$

The mixing time of a Markov chain MC is

$$\tau_{mix} = \min_n\{ \mid \mu_0 P^n - \mu \mid_{TV} \leq \varepsilon, \quad \forall \mu_0\}$$

Convergence study

There is a huge literature on convergence analysis, most of these are pretty much irrelevant for us in practice. Here we introduce a few measures.

The TV-norm is

$$|\mu_n - \mu|_{TV} = \frac{1}{2}\sum_{i\in\Omega}|\mu_n(i) - \mu(i)| = \sup_{A}|\mu_n(A) - \mu(A)|$$

$$|\nu_1 P - \nu_2 P|_{TV} \le C(P)\,|\nu_1 - \nu_2|$$

$$C(P) = \frac{1}{2}\max_{x,y}|P(x,\bullet) - P(y,\bullet)|_{TV}$$

$$KL(\mu \,||\, \nu P) \le KL(\mu \,||\, \nu)$$

Positive Recurrent

A state i is is said to be a recurrent state if it has $p(\tau_{ret}(i) < \infty) = 1$
Otherwise it is a transient state.

Furthermore, if $E[\tau_{ret}(i)] < \infty$
Then it is called a positively recurrent state,
otherwise it is a null-recurrent state.

Usually, the positive recurrence is a condition for spaces with infinite states.

Ergodicity theorem

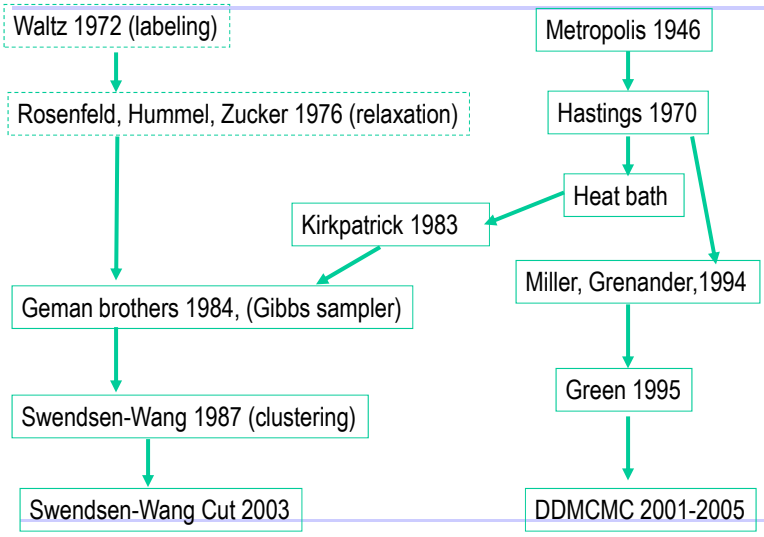
For an irreducible, positive recurrent Markov chain with stationary probability μ, in a state space Ω, let f(x) be any real vauled function with finite mean with respect to μ, then for any initial probability, almost surely we have

$$\lim_{n\to\infty} \frac{1}{N}\sum_{i=1}^{N} f(x_i) = \sum_{x\in\Omega} f(x)\mu(x) = E_{\mu}[f(x)]$$

To summarize, we have the following conditions for the Markov kernel K to be ergogic

0: stochastic --- each row sums to 1.
1: irreducible --- has 1 communication class
2: aperiodic --- any power of K has 1 communication class
3: globally balanced
4: positive recurrent

Some MCMC developments related to vision



Special cases

When the underlying graph G is a chain structure, then things are much simpler and many algorithms become equivalent.

Dynamic programming (Bellman 1957)
= Gibbs sampler (Geman and Geman 1984)
= Belief propagation (Pearl, 1985)
= exact sampling
= Viterbi (HMM 1967)