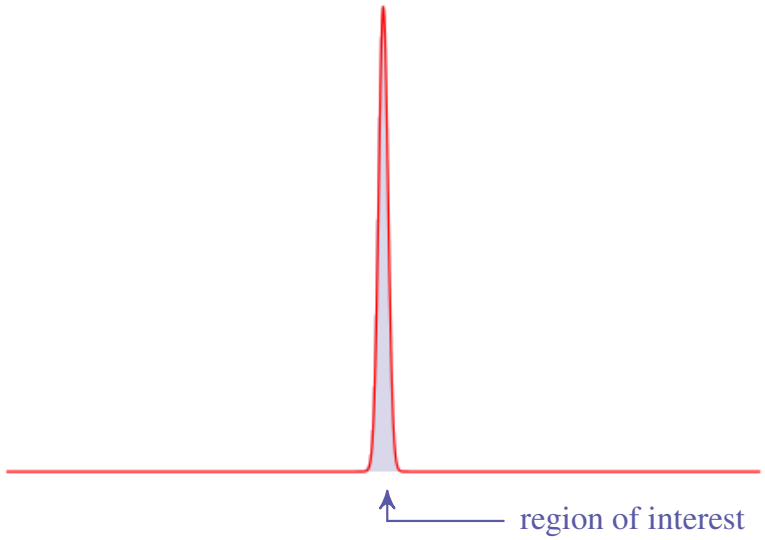
MARKOV CHAIN MONTE CARLO

MOTIVATION

Suppose we rejection-sample a distribution like this:



Once we have drawn a sample in the narrow region of interest, we would like to continue drawing samples within the same region. That is only possible if each sample *depends on the location of the previous sample*.

Proposals in rejection sampling are i.i.d. Hence, once we have found the region where p concentrates, we forget about it for the next sample.

Recall: Markov chain

- A sufficiently nice Markov chain (MC) has an invariant distribution P_{inv}.
- Once the MC has converged to P_{inv}, each sample X_t from the chain has marginal distribution P_{inv}.

Markov chain Monte Carlo

We want to sample from a distribution with density p. Suppose we can define a MC with invariant distribution $P_{\text{inv}} \equiv p$. If we sample $X_1, X_2, \ldots$ from the chain, then once it has converged, we obtain samples

$$X_t \sim p \ .$$

This sampling technique is called **Markov chain Monte Carlo** (**MCMC**).

Note: For a Markov chain, X_{t+1} can depend on X_t, so at least in principle, it is possible for an MCMC sampler to "remember" the previous step and remain in a high-probability location.

The Markov chains we discussed so far had a finite state space $\mathcal{X}$. For MCMC, state space now has to be the domain of p, so we often need to work with continuous state spaces.

Continuous Markov chain

A continuous Markov chain is defined by an initial distribution P_{init} and conditional probability $\phi(y|x)$, the **transition probability** or **transition kernel**.

In the discrete case, $\phi(y = j|x = i)$ is the entry Φ_{ij} of the transition matrix Φ.

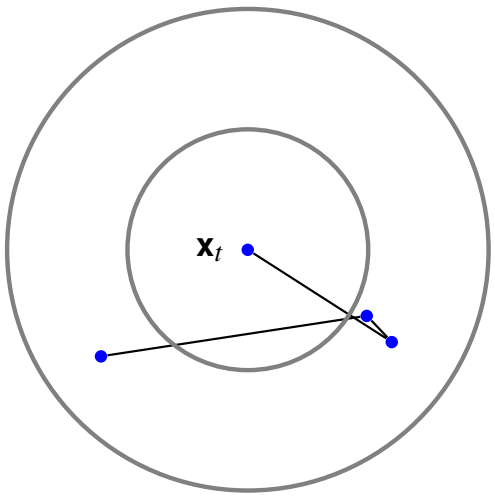
Example: A Markov chain on $\mathbb{R}^2$

We can define a very simple Markov chain by sampling

$$\mathbf{X}_{t+1}|\mathbf{X}_t = \mathbf{x}_t \quad \sim \quad g(\,.\,|\mathbf{x}_t, \sigma^2)$$

where $g(\mathbf{x}|\mu, \sigma^2)$ is a spherical Gaussian with fixed variance. In other words, the transition distribution is

$$\phi(\mathbf{x}_{t+1}|\mathbf{x}_t) := g(\mathbf{x}_{t+1}|\mathbf{x}_t, \sigma^2)\ .$$



A Gaussian (grey contours) is placed around the current point $\mathbf{x}_t$ to sample $\mathbf{X}_{t+1}$.

Recall: Finite case

- The invariant distribution $\mathbf{P}_{\text{inv}}$ is a distribution on the finite state space $\mathcal{X}$ of the MC (i.e. a vector of length $|\mathcal{X}|$).
- "Invariant" means that, if X_t is distributed according to $\mathbf{P}_{\text{inv}}$, and we execute a step $X_{t+1} \sim \phi(\,.\,|x_t)$ of the chain, then X_{t+1} again has distribution $\mathbf{P}_{\text{inv}}$.
- In terms of the transition matrix Φ:

$$\mathbf{P}_{\text{inv}} \cdot \Phi = \mathbf{P}_{\text{inv}}$$

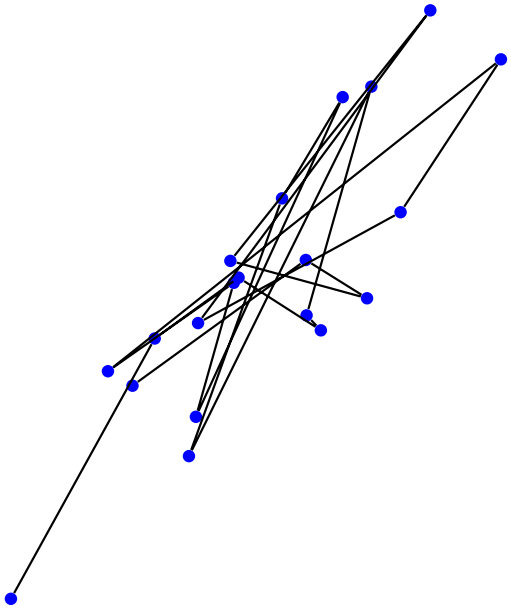
Continuous case

- $\mathcal{X}$ is now uncountable (e.g. $\mathcal{X} = \mathbb{R}^d$).
- The transition matrix Φ is substituted by the conditional probability ϕ.
- A distribution P_{inv} with density p_{inv} is invariant if

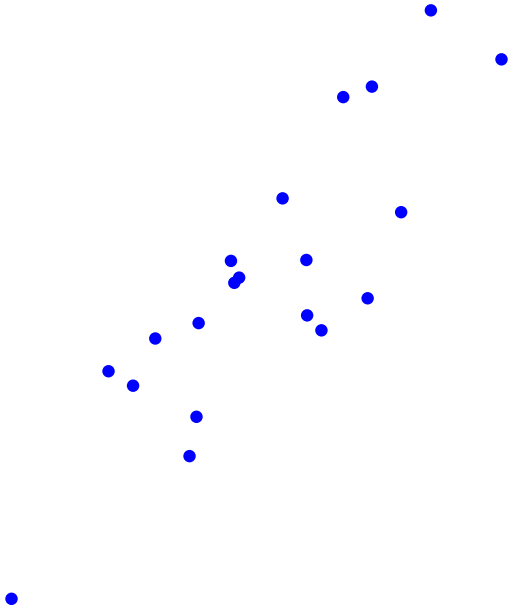
$$\int_{\mathcal{X}} \phi(y|x) p_{\text{inv}}(x) dx = p_{\text{inv}}(y)$$

  This is simply the continuous analogue of the equation $\sum_i \Phi_{ij} (\mathbf{P}_{\text{inv}})_i = (\mathbf{P}_{\text{inv}})_j$.

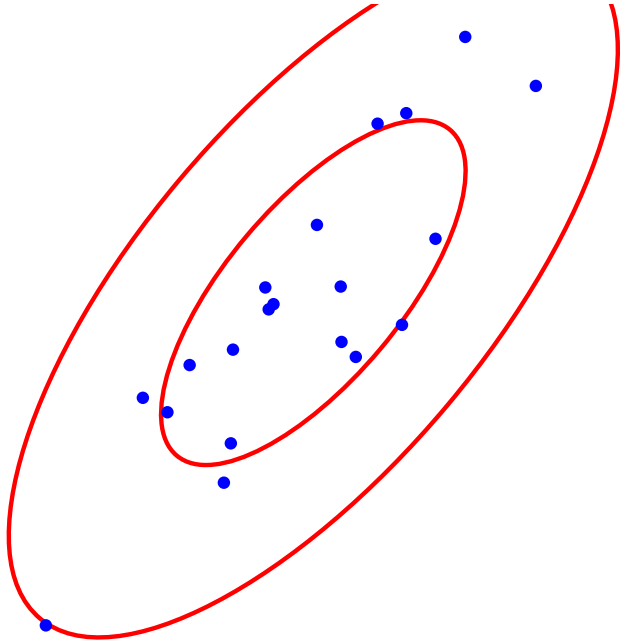
MARKOV CHAIN SAMPLING



We run the Markov chain n for steps. Each step moves from the current location x_t to a new x_{t+1}.



We "forget" the order and regard the locations $x_{1:n}$ as a random set of points.



If p (red contours) is both the invariant and initial distribution, each X_t is distributed as $X_t \sim p$.

Problems we need to solve

1. We have to construct a MC with invariant distribution p.
2. We cannot actually start sampling with $X_1 \sim p$; if we knew how to sample from p, all of this would be pointless.
3. Each point X_t is *marginally* distributed as $X_t \sim p$, but the points are *not* i.i.d.

Given is a continuous target distribution with density p.

Metropolis-Hastings (MH) kernel

1. We start by defining a conditional probability $q(y|x)$ on $\mathcal{X}$.
   q has nothing to do with p. We may e.g. choose $q(y|x) = g(y|x, \sigma^2)$, as in the previous example.
2. We define a **acceptance probability** (or density) A as

$$A(x_{t+1}|x_t) := \min\left\{1, \frac{q(x_t|x_{t+1})p(x_{t+1})}{q(x_{t+1}|x_t)p(x_t)}\right\}$$

   The normalization of p cancels in the quotient, so knowing $\tilde{p}$ is again enough.

3. We define the transition probability of the chain as

$$\phi(x_{t+1}|x_t) := q(x_{t+1}|x_t)A(x_{t+1}|x_t) + \delta_{x_t}(x_{t+1})c(x_t) \quad \text{where} \quad c(x_t) := \overbrace{\int q(y|x_t)(1-A(y|x_t))dy}^{\text{total probability that a proposal is sampled and then rejected}}$$

Sampling from the MH chain

At each step $t+1$, generate a proposal $X^* \sim q(\,.\,|x_t)$ and $U_t \sim \text{Uniform}[0,1]$.

- If $U_t \leq A(x^*|x_t)$, accept proposal: Set $x_{t+1} := x^*$.
- If $U_t > A(x^*|x_t)$, reject proposal: Set $x_{t+1} := x_t$.

The Metropolis-Hastings rejection kernel was defined as:

$$A(x_{t+1}|x_t) = \min\left\{1, \frac{q(x_t|x_{t+1})p(x_{t+1})}{q(x_{t+1}|x_t)p(x_t)}\right\} .$$

Hence, we certainly accept if the second term is larger than 1, i.e. if

$$q(x_t|x_{t+1})p(x_{t+1}) > q(x_{t+1}|x_t)p(x_t) .$$

For a symmetric proposal, this simplifies to $p(x_{t+1}) > p(x_t)$, which means:

- We always accept the proposal value x_{t+1} if it *increases* the probability under p.
- If it *decreases* the probability, we still accept with a probability which depends on the difference to the current probability.

Hill-climbing interpretation

- The MH sampler somewhat resembles a gradient ascent algorithm on p, which *tends* to move in the direction of increasing probability p.
- However:
  - The actual steps are chosen at random.
  - The sampler can move "downhill" with a certain probability.
  - When it reaches a local maximum, it does not get stuck there.

WHY p IS INVARIANT

Recall: Finite state space

For an irreducible, aperiodic chain on a finite $\mathcal{X}$, the invariant distribution (i) exists, (ii) is unique, and (iii) is obtained as the limit $\mathbf{P}_t \to \mathbf{P}_{\text{inv}}$, starting from any initial distribution $\mathbf{P}_{\text{init}}$.

Continuous state space

Here, irreducible and aperiodic do not guarantee that an invariant distribution exists.

Example: The chain $\mathbf{X}_{t+1} \sim g(\,.\,|\mathbf{x}_t, \sigma^2)$ on $\mathbb{R}^d$ is irreducible and aperiodic, but it keeps spreading out. The limit does not exist.

Detailed balance

We say that ϕ satisfies **detailed balance** with respect to p if $p(x)\phi(y|x) = p(y)\phi(x|y)$ for all x, y. Integrating over x gives

$$\int_{\mathcal{X}} \phi(y|x)p(x)dx = p(y) \int_{\mathcal{X}} \phi(x|y)dx = p(y) \ ,$$

so p is invariant. (Finite case: $P_i\Phi_{ij} = P_j\Phi_{ji}$ for all i, j implies $\mathbf{P} \cdot \Phi = \mathbf{P}$.)

The MH kernel satisfies detailed balance

For $y = x$ there is nothing to show. For $y \neq x$, only the accepted proposals contribute, and

$$p(x)\phi(y|x) = p(x)q(y|x)A(y|x) = \min\big\{p(x)q(y|x),\, p(y)q(x|y)\big\} \ ,$$

which is symmetric in x and y. Hence p **is invariant for the MH chain**.

Recall: Fundamental theorem on Markov chains

Suppose we sample $X_1 \sim P_{\text{init}}$ and $X_{t+1} \sim \phi(\,.\,|x_t)$. This defines a distribution P_t of X_t, which can change from step to step. If the Markov chain is nice (recall: irreducible and aperiodic), and if it has an invariant distribution, then

$$P_t \to P_{\text{inv}} \qquad \text{for} \qquad t \to \infty \ .$$

Note: Making precise what aperiodic means in a continuous state space is a bit more technical than in the finite case, but the theorem still holds. We will not worry about the details here.

Implication

- If we can show that $P_{\text{inv}} \equiv p$, we do not have to know how to sample from p.
- Instead, we can start with *any* P_{init}, and will get arbitrarily close to p for sufficiently large t.

The number m of steps required until $P_m \approx P_{\text{inv}} \equiv p$ is called the **mixing time** of the Markov chain. (In probability theory, there is a range of definitions for what exactly $P_m \approx P_{\text{inv}}$ means.)

In MC samplers, the first $m - 1$ samples are also called the **burn-in** phase. These are discarded for each run of the sampler:

$$\underbrace{X_1, \ldots, X_{m-1}}_{\text{Burn-in; discard.}}, \underbrace{X_m, X_{m+1}, \ldots}_{\text{Samples from (approximately) } p\text{; keep.}}$$

Convergence diagnostics

In practice, we do not know how large m is. There are a number of methods for assessing whether the sampler has mixed. Such heuristics are often referred to as **convergence diagnostics**.

Problem 2: Sequential Dependence

Even after burn-in, the samples from a MC are not i.i.d.

Strategy

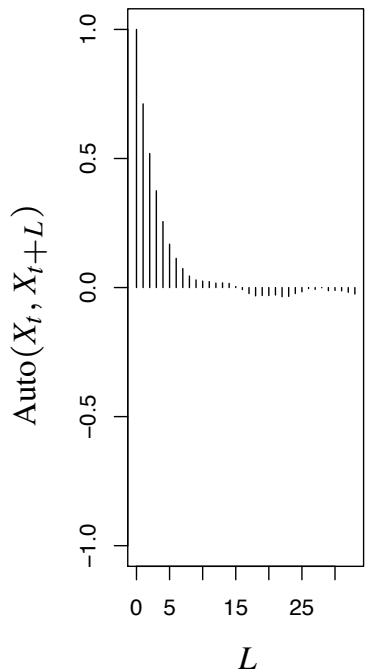
- Estimate empirically how many steps L are needed for x_t and x_{t+L} to be approximately independent. The number L is called the **lag**.
- After burn-in, keep only every Lth sample; discard samples in between.

Estimating the lag

The most commen method uses the **autocorrelation function**:

$$\text{Auto}(X_s, X_t) := \frac{\mathbb{E}[(X_s - \mu_s)(X_t - \mu_t)]}{\sigma_s \sigma_t} ,$$

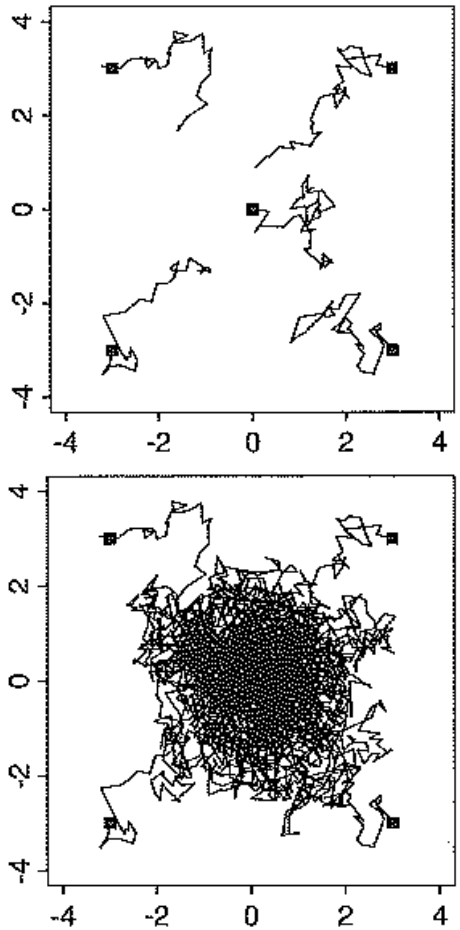
where μ_s is the mean and σ_s the standard deviation of X_s. We compute $\text{Auto}(X_t, X_{t+L})$ empirically from the sample for different values of L, and find the smallest L for which the autocorrelation is close to zero.



There are about half a dozen popular convergence crieteria; the one below is an example.

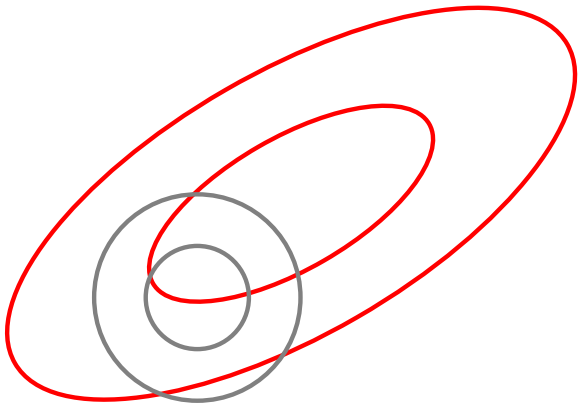
Gelman-Rubin criterion

- Start several chains at random. For each chain k, sample X_t^k has a marginal distribution P_t^k.
- The distributions of P_t^k will differ between chains in early stages.
- Once the chains have converged, all $P_t = P_{\text{inv}}$ are identical.
- One common criterion: Use a hypothesis test to compare P_t^k for different k (e.g. compare P_t^2 against null hypothesis P_t^1). Once the test does not reject anymore, assume that the chains are past burn-in.
- Altnernatively: In their original paper, Gelman and Rubin compare the variance within chains to the variance between chains and declare convergence when the two are sufficiently similar.



Reference: A. Gelman and D. B. Rubin: "Inference from Iterative Simulation Using Multiple Sequences", *Statistical Science*, 1992.

Everyone's favourite example: Two Gaussians



red = target distribution p
grey = proposal distribution q

- $\mathrm{Var}[q]$ too large:
  Will overstep p; many rejections.
- $\mathrm{Var}[q]$ too small:
  Many steps needed to achieve good coverage of domain.

If p is unimodal and can be roughly approximated by a Gaussian, $\mathrm{Var}[q]$ should be chosen as smallest covariance component of p.

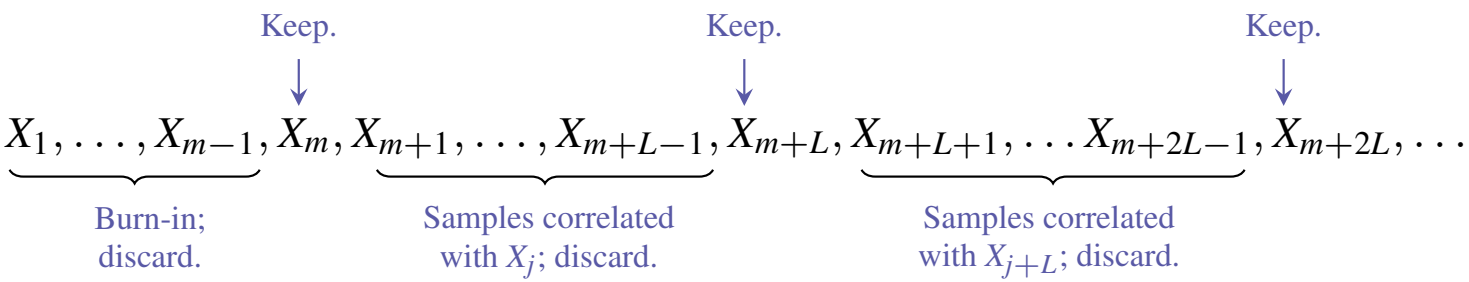
More generally

For complicated posteriors (recall: small regions of concentration, large low-probability regions in between) choosing q is much more difficult. To choose q with good performance, we already need to know something about the posterior.

There are many strategies, e.g. mixture proposals (with one component for large steps and one for small steps).

SUMMARY: MH SAMPLER

- MCMC samplers construct a MC with invariant distribution p.
- The MH kernel is one generic way to construct such a chain from p and a proposal distribution q.
- Formally, q does not depend on p (but arbitrary choice of q usually means bad performance).
- We have to discard an initial number of samples as burn-in to obtain samples (approximately) distributed according to p.
- After burn-in, we keep only every Lth sample (where $L = \text{lag}$) to make sure the x_t are (approximately) independent.



THE GIBBS SAMPLER

By far the most widely used MCMC algorithm is the Gibbs sampler.

Full conditionals

Suppose $\mathcal{L}(\mathbf{X})$ is a distribution on $\mathbb{R}^d$, so $\mathbf{X} = (X_1, \ldots, X_d)$. The conditional probability of the entry X_j given all other entries,

$$\mathcal{L}(X_j | X_1, \ldots, X_{j-1}, X_{j+1}, \ldots, X_d)$$

is called the **full conditional** distribution of X_j.

If $\mathbf{X}$ has density p, that means we are interested in a density

$$p(x_j | x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_d)$$

Gibbs sampling

- The Gibbs sampler is Markov chain Monte Carlo algorithm that generates each step dimension by dimension by sampling from the full conditionals.
- Gibbs sampling is only applicable if we can sample from the full conditionals for each dimension j.
- If so, it provides us with a *generic* way to derive a proposal distribution.

Proposal distribution

Suppose p is a density or mass function of a random vector with d entries. Given a random vector $\mathbf{X}_t$, we generate $\mathbf{X}_{t+1}$ coordinate-by-coordinate as follows:

$$\begin{aligned}
X_{t+1,1} &\sim p(\,.\,|X_{t,2}, \ldots, X_{t,d}) \\
&\vdots \\
X_{t+1,j} &\sim p(\,.\,|X_{t+1,1}, \ldots, X_{t+1,j-1}, X_{t,j+1}, \ldots, X_{t,d}) \\
&\vdots \\
X_{t+1,d} &\sim p(\,.\,|X_{t+1,1}, \ldots, X_{t+1,d-1})
\end{aligned}$$

Note well: Each new draw $X_{t+1,j}$ is used *immediately* to update the next dimension $j+1$.

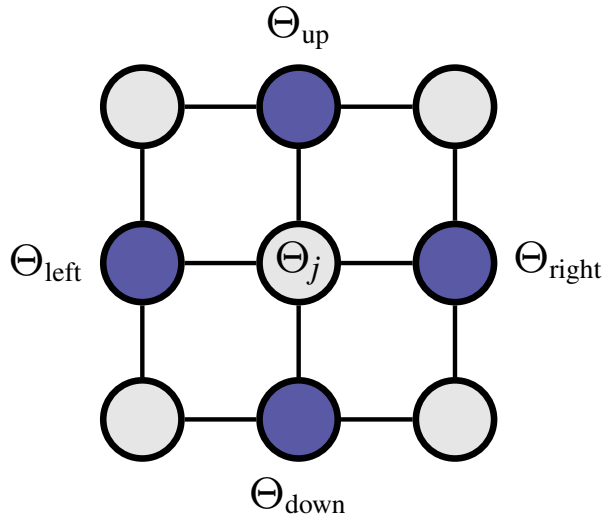
The MCMC algorithm that generates vectord $\mathbf{X}_1, \mathbf{X}_2, \ldots$ as above is called a **Gibbs sampler**.

Relationship to Metropolis-Hastings

- Updating the jth dimension is a Metropolis-Hastings step on the full vector $\mathbf{X}_t$, with the density $p(x_j|x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_d)$ of the full conditional as proposal density.
- One Gibbs step is the composition of these d Metropolis-Hastings steps.
- These MH samplers all have acceptance probability 1, so *proposals in Gibbs sampling are always accepted*.

EXAMPLE: MRF

In a MRF with d nodes, each dimension j corresponds to one vertex.



Full conditionals

In a grid with 4-neighbourhoods, for instance, the Markov property implies that

$$p(\theta_j|\theta_1,\ldots,\theta_{j-1},\theta_{j+1},\ldots,\theta_d) = p(\theta_j|\theta_{\text{left}},\theta_{\text{right}},\theta_{\text{up}},\theta_{\text{down}})$$

Specifically: Potts model with binary weights

Recall that, for sampling, knowing only $\tilde{p}$ (unnormalized) is sufficient:

$$\tilde{p}(\theta_j|\theta_1,\ldots,\theta_{j-1},\theta_{j+1},\ldots,\theta_d) =$$
$$\exp\Big(\beta(\mathbb{I}\{\theta_j=\theta_{\text{left}}\}+\mathbb{I}\{\theta_j=\theta_{\text{right}}\}+\mathbb{I}\{\theta_j=\theta_{\text{up}}\}+\mathbb{I}\{\theta_j=\theta_{\text{down}}\}\Big)$$

This is clearly very efficiently computable.

Sampling the Potts model

Each step of the sampler generates a sample

$$\theta_t = (\theta_{t,1}, \ldots, \theta_{t,d}) \ ,$$

where d is the number of vertices in the grid.

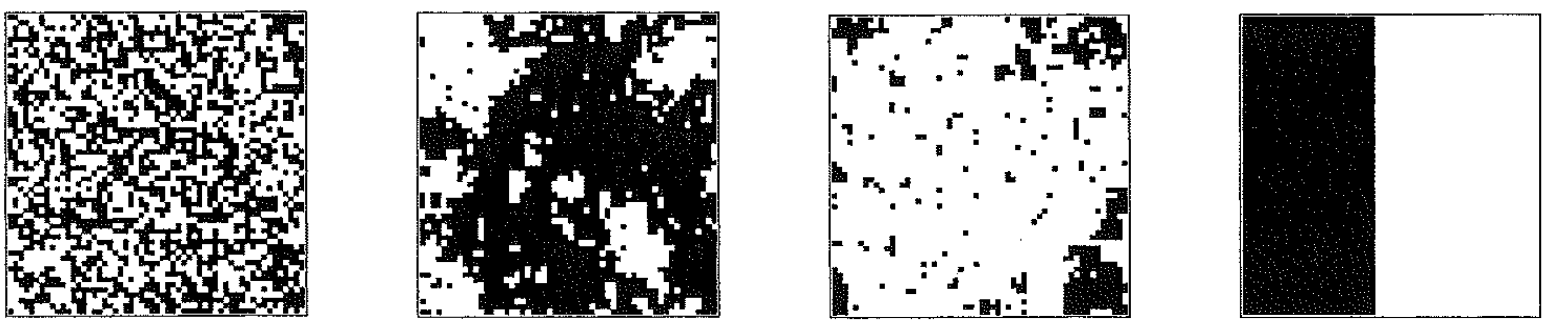
Gibbs sampler

Each step of the Gibbs sampler generates d updates according to

$$\begin{aligned}\theta_{t+1,\,j} &\sim \ p(\,.\,|\theta_{t+1,1}, \ldots, \theta_{t+1,j-1}, \theta_{t,j+1}, \ldots, \theta_{t,d}) \\ &\propto \ \exp\Big(\beta(\mathbb{I}\{\theta_{t+1,j} = \theta_{\text{left}}\} + \mathbb{I}\{\theta_{t+1,j} = \theta_{\text{right}}\} + \mathbb{I}\{\theta_{t+1,j} = \theta_{\text{up}}\} + \mathbb{I}\{\theta_{t+1,j} = \theta_{\text{down}}\})\Big)\end{aligned}$$

Recall the Ising model example

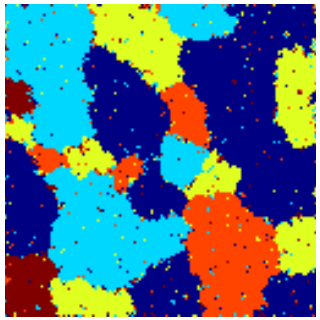
Samples from an Ising model on a 56×56 grid graph.



Increasing $\beta \quad \longrightarrow$

This example is due to Erik Sudderth (UC Irvine).

MRFs as "segmentation" priors

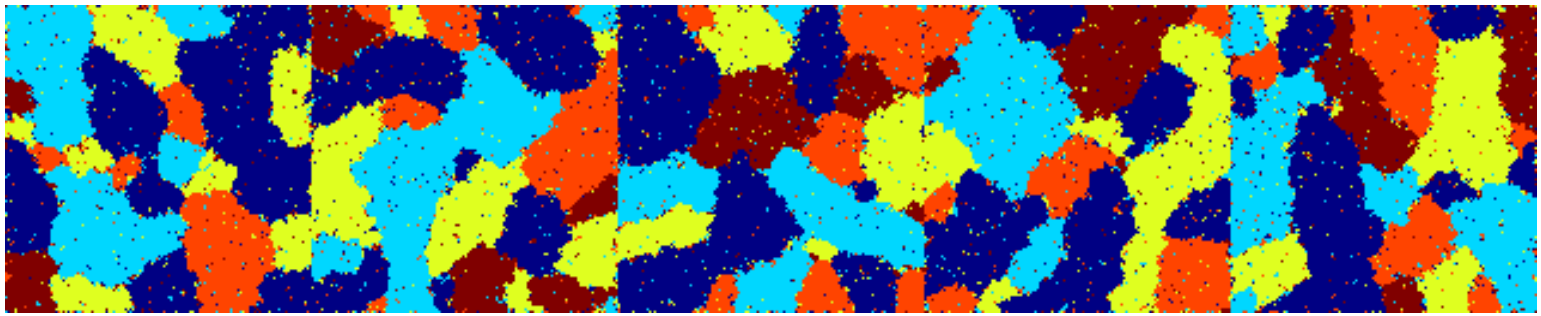


- MRFs where introduced as tools for image smoothing and segmentation by D. and S. Geman in 1984.
- They sampled from a Potts model with a Gibbs sampler, discarding 200 iterations as burn-in.
- Such a sample (after 200 steps) is shown above, for a Potts model in which each variable can take one out of 5 possible values.
- These patterns led computer vision researchers to conclude that MRFs are "natural" priors for image segmentation, since samples from the MRF resemble a segmented image.

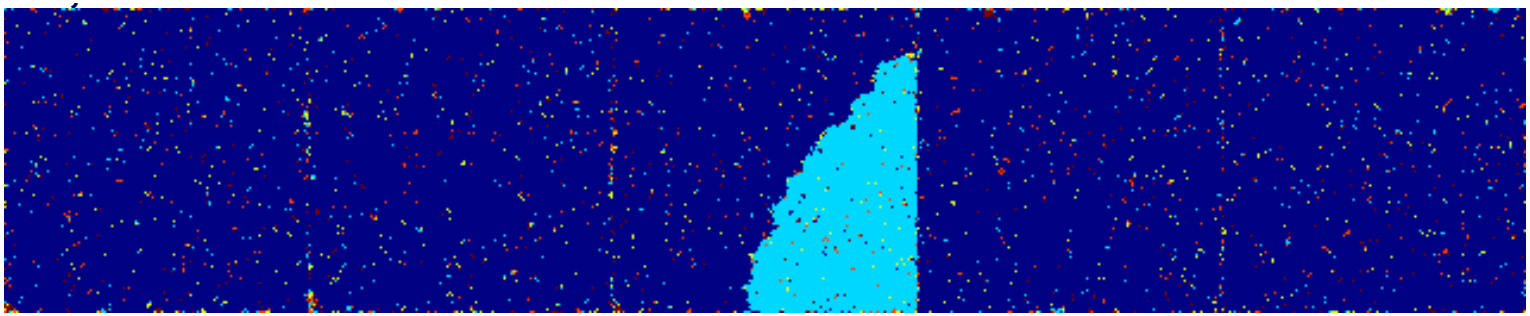
EXAMPLE: BURN-IN MATTERS

E. Sudderth and M. I. Jordan ran a Gibbs sampler on the same model, and recorded the state of the chain once after 200 iterations (as Geman & Geman had done), and again after 10000 iterations:

200 iterations



10000 iterations



Chain 1 Chain 5

- The "segmentation" patterns visible after 200 iterations are not sampled from the MRF distribution $p \equiv P_{\text{inv}}$, but rather from $P_{200} \neq P_{\text{inv}}$.
- The patterns do not show that MRFs are "natural" priors for segmentation problems, but simply *that the sampler's Markov chain is still in burn-in*.
- MRFs are smoothness priors, not segmentation priors.