# Bayesian Mixture Models and the Gibbs Sampler

David M. Blei
Columbia University

November 1, 2016

We have discussed probabilistic modeling, and have seen how the posterior distribution is the critical quantity for understanding data through a model.

The goal of probabilistic modeling is use domain and data-knowledge to build structured joint distributions, and then to reason about the domain (and exploit our knowledge) through the posterior and posterior predictive distributions.

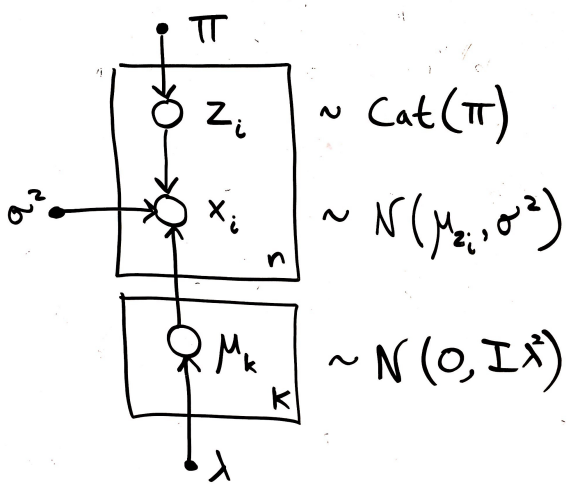
We have discussed tree propagation, a method for computing posterior marginals of any variables in a tree-shaped graphical model. In theory, if our graphical model was a tree, we could shade the observations and do useful inferences about the posterior.

For many interesting models, however, the posterior is not tractable to compute. Either the model is not a tree or the messages are not tractable to compute (because of the form of the potentials). Most modern applications of probabilistic modeling rely on *approximate posterior inference* algorithms.

We now discuss an important method for approximate posterior inference. In parallel, we begin to discuss the building blocks of complex models.

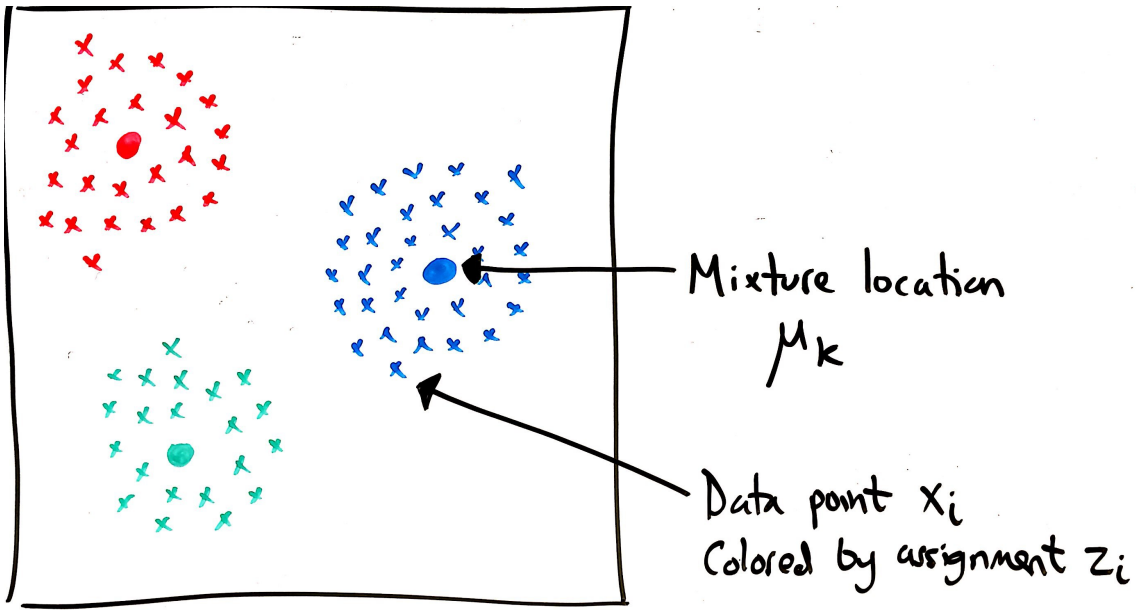
## Bayesian mixture of Gaussians

To lock ideas, and to give you a flavor of the simplest interesting probabilistic model, we will first discuss Bayesian mixture models. Here is the Bayesian mixture of Gaussians,




(We haven't yet discussed the multivariate Gaussian. But we don't need to yet. Here each mixture component is $\mu_k = [\mu_x, \mu_y]$ and we generate the $x$ and $y$ coordinates independently from their respective distributions.)

To get a feel for what this model is about, we generate data from it.

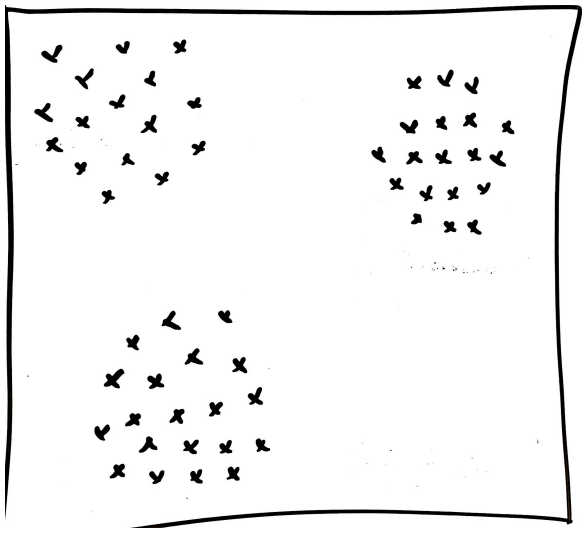


### The posterior distribution

The posterior distribution is a distribution over the latent variables, the cluster locations and the cluster assignments. Let $\mathbf{x} = x_{1:n}$, $\mathbf{z} = z_{1:n}$, and $\boldsymbol{\mu} = \mu_{1:K}$. The posterior is

$$p(\mathbf{z}, \boldsymbol{\mu} \mid \mathbf{x}). \tag{1}$$

This gives an understanding of the data (at least, a grouping into $K$ groups).



What is this posterior? The mixture model assumes that each data point came from one of $K$ distributions. However, it is unknown what those distributions are and how the data were assigned to them. The posterior is a conditional distribution over these quantities. Through the posterior, a mixture model tells us about a grouping of our data.

The posterior is proportional to the joint. Let's write down the log of the joint,

$$\log p(\boldsymbol{\mu}, \mathbf{x}, \mathbf{z}) = \log \sum_{k=1}^{K} \log p(\mu_k) + \sum_{i=1}^{n} \log p(z_i \mid \pi) + \log p(x_i \mid \boldsymbol{\mu}, z_i). \tag{2}$$

To gain intuition, assume that $n$ is very large and so the likelihood term dominates this expression. Also assume that the prior on the mixture assignments $\pi$ is fixed, so the term $\log p(z_i)$ does not play a role.

We can see that this expression is dominated by the likelihood $\log p(x_i \mid \boldsymbol{\mu}, z_i)$. This is equal to

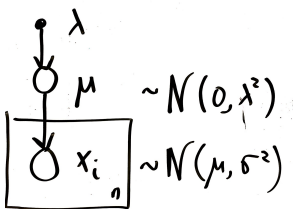
$$\log p(x_i \mid \boldsymbol{\mu}, z_i) = -\frac{1}{2\sigma^2}(x_i - \mu_{z_i})^2 + \text{const.} \tag{3}$$

In other words, the posterior seeks configurations of mixture locations $\mu_k$ and mixture assignments $z_i$ such that the data are close to their assigned locations. This is why mixtures of Gaussians partitions the data in a sensible way.

### The posterior is intractable to compute

We cannot compute the posterior exactly. Let's see why.

**Aside: The Gaussian is conjugate to the Gaussian.** Consider a simple model, where we draw a Gaussian mean $\mu$ from a Gaussian prior $\mathcal{N}(0, \lambda)$ and then generate $n$ data points from a Gaussian $\mathcal{N}(\mu, \sigma^2)$. (We fix the variance $\sigma^2$.) Here is the graphical model




You have seen the beta-Bernoulli; this is another example of a conjugate pair. Given $\mathbf{x}$ the posterior distribution of $\mu$ is $\mathcal{N}(\hat{\mu}, \hat{\lambda})$, where

$$\hat{\mu} = \left(\frac{n/\sigma^2}{n/\sigma^2 + 1/\lambda^2}\right)\bar{x} \tag{4}$$

$$\hat{\lambda} = \left(n/\sigma^2 + 1/\lambda^2\right)^{-1}, \tag{5}$$

where $\bar{x}$ is the sample mean.

Just as for the beta-Bernoulli, as $n$ increases the posterior mean approaches the sample mean and the posterior variance approaches zero. (Note: this is the posterior mean and variance of the unknown *mean*. The data variance $\sigma^2$ is held fixed in this analysis.)

**Back to the mixture of Gaussians.** But now suppose we are working with a mixture of Gaussians. In that case, $p(\boldsymbol{\mu} \mid \mathbf{x})$ is not easy. Suppose the prior proportions $\pi$ are fixed and $K = 3$. Consider the posterior of the mixture components,

$$p(\mu_1, \mu_2, \mu_3 \mid \mathbf{x}) = \frac{p(\mu_1, \mu_2, \mu_3, \mathbf{x})}{\int\int\int p(\mu_1', \mu_2', \mu_3', \mathbf{x}) d\mu_1' d\mu_2' d\mu_3'}. \tag{6}$$

Note this implicitly marginalizes out the mixture assignments.

The numerator is easy,

$$\text{numerator} = p(\mu_1)p(\mu_2)p(\mu_3)\prod_{i=1}^{n} p(x_i \mid \mu_1, \mu_2, \mu_3) \tag{7}$$

where each likelihood term marginalizes out the $z_i$ variable,

$$p(x_i \mid \mu_1, \mu_2, \mu_3) = \sum_{k=1}^{K} \pi_k \, p(x_i \mid \mu_k). \tag{8}$$

But consider the denominator, which is the marginal probability of the data,

$$p(\mathbf{x}) = \int\int\int p(\mu_1')p(\mu_2')p(\mu_3')\prod_{i=1}^{n}\sum_{k=1}^{K} \pi_k \, p(x_i \mid \mu_k') d\mu_1' d\mu_2' d\mu_3'. \tag{9}$$

This integral is difficult to compute. One way to see this is to notice (or be told that) it is a type of hypergeometric function.

A more direct way to see that it's difficult is to bring the summation to the outside of the integral

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{z}) \int\int\int p(\mu_1')p(\mu_2')p(\mu_3')\prod_{i=1}^{n} p(x_i \mid \mu_{z_i}') d\mu_1' d\mu_2' d\mu_3'. \tag{10}$$

This can be decomposed by partitioning the data according to $\mathbf{z}$,

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{z}) \prod_{k=1}^{3} \left( \int p(\mu_k') \prod_{\{i : z_i = k\}} p(x_i \mid \mu_k') d\mu_k' \right). \tag{11}$$

Each term in the product is an integral under the conjugate prior, which is an expression we can compute. (We will see that later on.) But there are $3^n$ different

assignments of the data to consider.

To use a Bayesian mixtures of Gaussians (and many other models), we need approximate inference.

### The Gibbs sampler for a mixture of Gaussians

Gibbs sampling (and all of Markov chain Monte Carlo) approximates a distribution with a set of samples. For example, in the mixture model,

$$p(\boldsymbol{\mu}, \mathbf{z} \mid \mathbf{x}) \approx \frac{1}{B} \sum_{b=1}^{B} \delta_{(\boldsymbol{\mu}^{(b)}, \mathbf{z}^{(b)})}(\boldsymbol{\mu}, \mathbf{z}). \tag{12}$$

We first discuss Gibbs sampling for mixtures of Gaussians. Later we see how it generalizes to other models.

The Gibbs is an iterative algorithm. It maintains a value for each latent variable. In each iteration, it samples from each latent variable conditional on the other latent variables and the observations. Call each of these distributions a *complete conditional*.

See Figure 1 for Gibbs sampling for Gaussian mixtures. This sampler iterates between probabilistically assigning each data point to a mixture component and probabilistically updating the location of each mixture component. Notice that when we sample a variable its value changes in what we subsequently condition on. E.g., when we sample $\mu_1$, this changes what $\mu_1$ is for the subsequent samples.

The theory around Gibbs sampling says that if we do this many times, the resulting sample will be a sample from the exact posterior. The reason is that we have defined a Markov chain whose state space are the latent variables and whose *stationary distribution* is the posterior we care about. After many iterations $t$, the marginal distribution of $\boldsymbol{\mu}^{(t)}$ and $\mathbf{z}^{(t)}$ is the exact posterior. Doing this multiple times, we can obtain $B$ samples from the posterior.

**Input:** data $\mathbf{x}$ and a number of components $K$.
Initialize mixture locations $\boldsymbol{\mu}$ randomly.

Maintain mixture locations $\boldsymbol{\mu}$ and mixture assignments $\mathbf{z}$.
**repeat**
  **for** *each data point* $i$ **do**
    Sample $z_i \mid \{\boldsymbol{\mu}, \mathbf{z}_{-i}, \mathbf{x}\}$ (Equation 15)
  **end**
  **for** *each mixture component* $k$ **do**
    Sample $\mu_k \mid \{\boldsymbol{\mu}_{-k}, \mathbf{z}, \mathbf{x}\}$ (Equation 17)
  **end**
**until**

**Figure 1:** The Gibbs sampler for mixture of Gaussians.

.

**Details about the complete conditionals**

Let's work out each step of the algorithm, beginning with the complete conditional of $z_i$. First, the graphical model implies a conditional independence,

$$p(z_i \mid \boldsymbol{\mu}, z_{-i}, \mathbf{x}) = p(z_i \mid \boldsymbol{\mu}, x_i). \tag{13}$$

Now calculate the distribution,

$$p(z_i \mid \boldsymbol{\mu}, x_i) \propto p(z_i)p(x_i \mid \mu_{z_i}) \tag{14}$$

$$= \pi_{z_i}\phi(x_i; \mu_{z_i}, \sigma^2). \tag{15}$$

What is this? To keep things simple, assume $\pi_k = 1/K$. Then this is a categorical distribution where the probability of the the $k$th is proportional to the likelihood of the $i$th data point under the $k$th cluster.

Notes:

- Categorical distributions are easy to sample from.
- This distribution requires that we know $\boldsymbol{\mu}$.

Now derive the complete conditional of $\mu_k$. Again, observe a conditional indepen-

dence from the graphical model,

$$p(\mu_k \mid \mu_{-k}, \mathbf{z}, \mathbf{x}) = p(\mu_k \mid \mathbf{z}, \mathbf{x}). \tag{16}$$

We calculate the distribution intuitively. If we know the cluster assignments, what is the conditional distribution of $\mu_k$? It is simply a posterior Gaussian, conditional on the data that were assigned to the $k$th cluster.

Technically: Let $z_i$ be an indicator vector, a $K$-vector with a single one. Then,

$$\mu_k \mid \mathbf{z}, \mathbf{x} \sim \mathcal{N}(\hat{\mu}_k, \hat{\lambda}_k^2) \tag{17}$$

. Define

$$n_k = \sum_{i=1}^{n} z_i^k \tag{18}$$

$$\bar{x}_k = \frac{\sum_{i=1}^{n} z_i^k x_i}{n_k}. \tag{19}$$

and

$$\hat{\mu}_k = \left( \frac{n_k/\sigma^2}{n_k/\sigma^2 + 1/\lambda^2} \right) \bar{x}_k \tag{20}$$

$$\hat{\lambda}^2 = \left( n_k/\sigma^2 + 1/\lambda^2 \right)^{-1}. \tag{21}$$

**Important**: Conjugacy helps us, even when we cannot compute the posterior.

This is an approximate inference algorithm for mixtures of Gaussians. At each iteration, we first sample each mixture assignment from Equation 15 and then sample each mixture location from Equation 17.

Discussion and practicalities:

- This sampler results in one sample from the posterior. To get $B$ samples, we run several times. In practice, we begin from an *initial state* and run for a fixed number of *burn in* iterations. We then continue to run the algorithm, collecting samples a specified *lag*.

  Initialization, burn-in, and lag are important practical issues. There are no good

principled solutions, but many ad-hoc ones work well. (We discuss initialization a little bit below.)

- Notice that conditional independencies in the complete conditionals give us opportunities to parallelize. What can be parallelized here?

- Gibbs sampling closely relates to the expectation-maximization (EM) algorithm.

  In the EM algorithm we iterate between the E-step and M-step. In the E-step we compute the conditional distribution of each assignment given the locations. This is precisely Equation 15.

  In the M-step we update the locations at maximum likelihood estimates under expected sufficient statistics. As we know, the MLE relates to the Bayesian posterior in Equation 20.

- The theory implies that we need infinite lag time and infinite burn-in. Practical decisions around Gibbs sampling can be difficult to make. (But, happily, in practice it's easy to come up with sensible unjustified choices.) One quantity to monitor is

$$\log p(\boldsymbol{\mu}^{(t)}, \mathbf{z}^{(t)}, \mathbf{x}) = \sum_{k=1}^{K} \log p(\mu_k^{(t)}) + \sum_{i=1}^{n} \log p(z_i^{(t)}) + \log p(x_i \mid z_t^{(t)}, \boldsymbol{\mu}^{(t)}). \tag{22}$$

  This is the log joint of the assignments of the latent variables and observations.

  This further relates to EM, which optimizes the conditional expectation (over the mixture assignments $z$) of this quantity.

- The perfomance of the algorithm is sensitive to initialization. The best practical advice is to initialize "mildly," in a way that does not lead to confident estimates of various latent variables. Here, for example, you might center the mixture components at some random perturbation away from zero. (This assumes you have centered the data.)

### The collapsed Gibbs sampler

Sometimes we can integrate out hidden random variables from a complete conditional. This is called *collapsing*.

**Input:** data $\mathbf{x}$ and a number of components $K$.
Initialize mixture assignments $\mathbf{z}$ randomly.

Maintain mixture assignments $\mathbf{z}$ and two derived quantities

$$n_k \triangleq \sum_{i=1}^{n} z_i^k \qquad \text{(number of items per cluster)}$$
$$s_k \triangleq \sum_{i=1}^{n} z_i^k x_i \qquad \text{(cluster sum)}$$

**repeat**
  **for** *each data point* $i$ **do**
    "Knock out" $x_i$ from its currently assigned cluster $z_i$.

$$n_{z_i} \leftarrow n_{z_i} - 1$$
$$s_{z_i} \leftarrow s_{z_i} - x_i$$

    Use $n_k$ and $s_k$ to sample $z_i$ from Equation 23.
  **end**
**until**

**Figure 2:** Collapsed Gibbs sampling for Gaussian mixtures.

It is a good idea because it usually leads to faster convergence of the Markov chain to the stationary distribution. (We discuss these concepts below.) But it is also usually more costly per iteration.

In the mixture of Gaussians, consider collapsing the mixture locations,

$$p(z_i = k \mid z_{-i}, \mathbf{x}) \propto p(z_i = k)\, p(x_i \mid z_{-i}, x_{-i}, z_i = k). \tag{23}$$

The second term is simply a posterior predictive distribution

$$p(x_i \mid z_{-i}, x_{-i}, z_i) = \int_{\mu_k} p(x_i \mid \mu_k)\, p(\mu_k \mid z_{-i}, x_{-i}). \tag{24}$$

This is like immediately updating the mixture components after resampling the mixture assignment. Figure 2 shows collapsed Gibbs sampling for Gaussian mixtures.

Collapsed Gibbs sampling can be more expensive at each iteration, but converges faster. Typically, if you can collapse then it is worth it.

**The posterior predictive (optional)**

As usual, the posterior also gives a posterior predictive distribution,

$$p(x_{n+1} \mid \mathbf{x}) = \int p(x_{n+1}, \boldsymbol{\mu} \mid \mathbf{x}) \mathrm{d}\boldsymbol{\mu} \tag{25}$$

$$= \int p(x_{n+1} \mid \boldsymbol{\mu}, \mathbf{x}) p(\boldsymbol{\mu} \mid \mathbf{x}) \mathrm{d}\boldsymbol{\mu} \tag{26}$$

$$= \int p(x_{n+1} \mid \boldsymbol{\mu}) p(\boldsymbol{\mu} \mid \mathbf{x}) \mathrm{d}\boldsymbol{\mu} \tag{27}$$

$$= \mathbb{E}\left[p(x_{n+1} \mid \boldsymbol{\mu}) \mid \mathbf{x}\right]. \tag{28}$$

Note we removed $\mathbf{x}$ from the RHS of the conditional distribution. This is thanks to conditional independence. From Bayes ball we have that

$$x_{n+1} \perp\!\!\!\perp \mathbf{x} \mid \boldsymbol{\mu} \tag{29}$$

In more detail, the posterior predictive distribution is

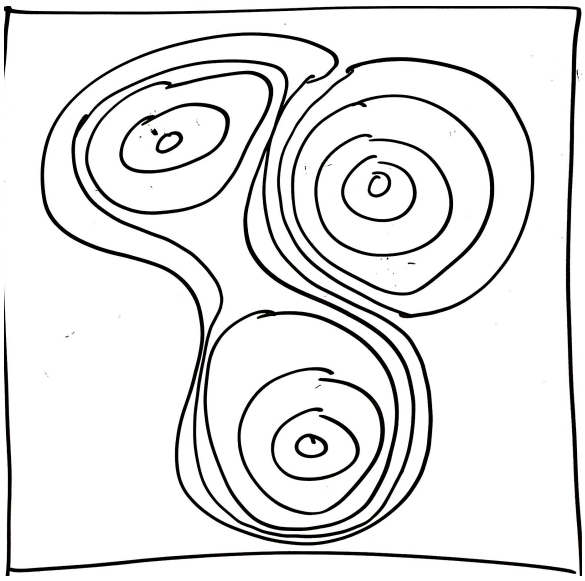
$$p(x_{n+1} \mid \mathbf{x}) = \int_{\boldsymbol{\mu}} p(x_{n+1} \mid \boldsymbol{\mu}) p(\boldsymbol{\mu} \mid \mathbf{x}) d\boldsymbol{\mu} \tag{30}$$

$$= \int_{\boldsymbol{\mu}} \left( \sum_{k=1}^{K} p(z_{n+1} = k) p(x_{n+1} \mid \mu_k) \right) d\boldsymbol{\mu} \tag{31}$$

$$= \sum_{k=1}^{K} p(z_{n+1} = k) \left( \int_{\mu_k} p(x_{n+1} \mid \mu_k) p(\mu_k \mid \mathbf{x}) d\mu_k \right) \tag{32}$$

What is this? We consider $x_{n+1}$ as coming from each of the possible mixture locations (one through $K$) and then take a weighted average of its posterior density at each.

This is a multi-modal distribution over the next data point. Here is a picture:



This predictive distribution involves the posterior through $p(\mu_k \mid \mathbf{x})$, the posterior distribution of the $k$th component given the data.

Contrast this with the predictive distribution we might obtain if we used a single Gaussian to model the data. In that case, the mean will be at a place where there is little data.

## Gibbs sampling

Gibbs sampling constructs a Markov chain on the latent variables, defined by the transition $p_{\text{gibbs}}(\mathbf{z}_{t+1} \mid \mathbf{z}_t)$. This transition implements what we described above, iteratively sampling each latent variable conditional on all the others and the observations. (Notice the observations don't appear in the transition probabilities, because they are implicitly part of the definition of the chain.)

Define the $t$-th marginal distribution to be the distribution that marginalizes out $\mathbf{z}_1, \ldots, \mathbf{z}_{(t-1)}$,

$$p^{(t)}_{\text{gibbs}}(\mathbf{z}_t) = \sum_{\mathbf{z}_{1:(t-1)}} p(\mathbf{z}_1, \ldots, \mathbf{z}_{(t-1)}, \mathbf{z}_t). \tag{33}$$

Under certain conditions, a Markov chain "converges" in that the distribution of the long-run marginals of the variables do not change. Suppose a Markov chain has

converged to a distribution $\pi(\mathbf{z})$. Consider drawing from this distribution and then taking one step in the chain. Then

$$\sum_{\mathbf{z}'} \pi(\mathbf{z}') p_{\text{gibbs}}(\mathbf{z} \mid \mathbf{z}') = \pi(\mathbf{z}) \tag{34}$$

For obvious reasons $\pi(\cdot)$ is called the *stationary distribution*. The Gibbs sampler defines a chain whose stationary distribution is $p(\mathbf{z} \mid \mathbf{x})$.

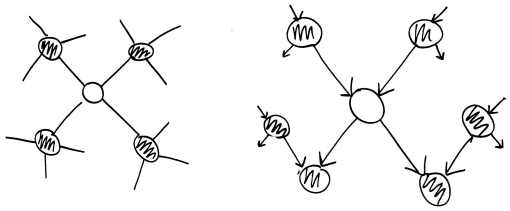
The ease of implementing a Gibbs sampler depends on how easy it is to compute and sample from the various complete conditionals. In a graphical model, the complete conditional depends on the *Markov blanket* of the node.

Suppose the nodes are $x_1, \ldots, x_k$ (observed and unobserved). In an undirected graphical model, the complete conditional only depends on a node's neighbors,

$$p(x_i \mid x_{-i}) = p(x_i \mid x_{\mathcal{N}(i)}). \tag{35}$$

In a directed model, the complete conditional depends on a node's parents, children, and other parents of its children,

We can see these facts from the graphical model and separation / d-separation



Again, notice the theme. Difficult global computation is made easier by many local computations. Gibbs sampling is a form of "message passing."

### A loose history of MCMC

Metropolis et al. (1953) introduces the Metropolis algorithm in the context of some integrals that come up in physics. (He also invented simulated annealing in the same paper.) This work stemmed from his work in the 40s (with others) in Los Alamos.

Hastings (1970) generalizes the algorithm to Metropolis-Hastings and sets it in a more statistical context. He shows that the Metropolis algorithm (and Metropolis-Hastings) works because they sample from a Markov chain with the appropriate stationary distribution.

Geman and Geman (1984) developed the Gibbs sampler for Ising models, showing that it too samples from an appropriate Markov chain. Gelfand and Smith (1990) built on this work to show how Gibbs sampling can be used in many Bayesian settings. This is also what we've shown in these notes.[1]

In the 90s, statisticians like Tierney, Kass, and Gelman (in a series of papers and books) solidified the relationship between Metropolis, Metropolis-Hastings, and Gibbs sampling. In parallel, the rise of computing power transformed where and how these algorithms can be used.

Today, MCMC is a vital tool for modern applications of probabilistic models.

## References


Gelfand, A. and Smith, A. (1990). Sampling based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85:398–409.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741.

Hastings, W. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, M., and Teller, E. (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092.


[1] At that same time, Gelfand (who was faculty at the University of Connecticut) played a monthly poker game with a group of mathematicians and statisticians including Prof. Ron Blei (relation to author: father). They called it "the probability seminar." Sometimes they played in the author's childhood home. Early in the evening, he was allowed to grab a handful of pretzels but, otherwise, was encouraged to stay out of the way.