STAT 339
Markov Chain Monte Carlo (MCMC)

7 April 2017

Some theory and intuition about MCMC generally

Outline

Gibbs Sampling in Mixture Models
- Sampling as an Approximation Method
- Gibbs Sampling Intuition and Motivation
- Concretely: The Algorithm

A Bit of MCMC Theory

Outline

Gibbs Sampling in Mixture Models
- Sampling as an Approximation Method
- Gibbs Sampling Intuition and Motivation
- Concretely: The Algorithm

A Bit of MCMC Theory

Outline

Gibbs Sampling in Mixture Models
- Sampling as an Approximation Method
- Gibbs Sampling Intuition and Motivation
- Concretely: The Algorithm

A Bit of MCMC Theory

Intractable Posterior

- It is easy to write down the unnormalized posterior density:
$$p(\boldsymbol{\theta} \mid \mathbf{Y}) \propto p(\boldsymbol{\theta}) p(\mathbf{Y} \mid \boldsymbol{\theta})$$
- GMM case:
$$p(\boldsymbol{\theta} \mid \mathbf{Y}) \propto p(\boldsymbol{\pi}) \prod_{k=1}^{K} p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \prod_{n=1}^{N} \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{y}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right)$$
where, e.g., $p(\boldsymbol{\pi})$ is Dirichlet, $p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is Normal-Inverse Wishart (or $p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1})$ is Normal-Wishart)
- But hard to do much with it (except maybe find a local maximum)
- Solution: Try to draw *samples* from it.

When can we sample from but not analyze a distribution?

- Consider trying to evaluate $P(\text{Straight})$ in poker (say)
- You might be able to do this analytically, but if you have a computer handy, it is easy to simulate dealing out several million poker hands
- Then, simply count:

$$P(\text{Straight}) \approx \frac{1}{S} \sum_{s=1}^{S} I(\text{Did I get a straight in hand s?})$$

Outline

Gibbs Sampling in Mixture Models
- Sampling as an Approximation Method
- Gibbs Sampling Intuition and Motivation
- Concretely: The Algorithm

A Bit of MCMC Theory

Gibbs Sampling as "Stochastic EM"

- The EM algorithm involves, iteratively
  1. Computing an expectation over cluster assignments (using the posterior, conditioned on parameter values)
  2. Arg-Maximizing parameter values (using the likelihood/posterior conditioned on expected cluster assignments)
- Gibbs sampling (in this context) involves, iteratively
  1. Sampling cluster assignments (using the posterior, conditioned on parameter values)
  2. Sampling parameter values (using the posterior, conditioned on cluster assignments)

What does Gibbs Yield?

- Over many iterations, Gibbs sampling yields a collection of many cluster assignments and many parameter values, distributed according to the joint posterior

$$\{\{z_n^{(s)}\}_{n=1}^N, \boldsymbol{\pi}^{(s)}, \{\boldsymbol{\mu}_k^{(s)}, \boldsymbol{\Sigma}_k^{(s)}\}_{k=1}^K\}_{s=1}^S$$

- These can be used to approximate more or less any function of the assignments and parameters we want
- Example:

$$p(z_n = z_m \mid \mathbf{Y}) \approx \frac{1}{S}\sum_{s=1}^{S} I(z_n^{(s)} = z_m^{(s)})$$

- Or, for future data:

$$p(z_{new} = z_m \mid \mathbf{y}_{new}, \mathbf{Y}) \approx \frac{1}{S}\sum_{s=1}^{S} \frac{\pi_{z_m^{(s)}}^{(s)} \mathcal{N}(\mathbf{y}_{new} \mid \boldsymbol{\mu}_{z_m^{(s)}}, \boldsymbol{\Sigma}_{z_m^{(s)}})}{\sum_{k=1}^K \pi_k^{(s)} \mathcal{N}(\mathbf{y}_{new} \mid \boldsymbol{\mu}_k^{(s)}, \boldsymbol{\Sigma}_k^{(s)})}$$

Outline

Gibbs Sampling in Mixture Models
- Sampling as an Approximation Method
- Gibbs Sampling Intuition and Motivation
- Concretely: The Algorithm

A Bit of MCMC Theory

Concretely: The Algorithm

- In general, Gibbs sampling applies when we have multiple sets of unknown parameters, but where it is difficult to sample all of them directly from their joint distribution.
- Mixture model: params are $\mathbf{z}$, $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$
- Idea: partition the parameters into "blocks", and iteratively fix all but one block and sample the remaining from its posterior conditioned on the others
- Here, we need

$$
\begin{aligned}
&1\text{: } p(\mathbf{z} \mid \mathbf{Y}, \boldsymbol{\theta}, \boldsymbol{\pi}) \\
&2\text{: } p(\boldsymbol{\theta}, \boldsymbol{\pi} \mid \mathbf{z}, \mathbf{Y})
\end{aligned}
$$

Summary: Gibbs Algorithm for GMM

1. Initialize cluster assignments, $\mathbf{z}^{(0)}$ (e.g., using K-means)
2. For $s = 1, \ldots, S$, sample
   (a) $\boldsymbol{\mu}_k^{(s)}, \boldsymbol{\Sigma}_k^{(s)} \sim p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \mid \mathbf{z}^{(s-1)}, \mathbf{Y}), k = 1, \ldots, K$
   (or, alternatively, subdivide into two blocks
   with $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ sampled separately given the other)
   (b) $\boldsymbol{\pi}^{(s)} \sim p(\boldsymbol{\pi} \mid \mathbf{z}^{(s-1)})$
   (c) $z_n^{(s)} \sim p(z_n \mid \boldsymbol{\pi}^{(s)}, \boldsymbol{\theta}^{(s)}, \mathbf{y}_n), n = 1, \ldots, N$

Specific Conditional Distributions

Assuming conjugate priors and Σ_k restricted to diagonal w/ elements $h_{k,d} = \sigma^2_{k,d}$

$$p(\mu_{k,d} \mid h_{k,d}, \mathbf{z}, \mathbf{Y}) = \mathcal{N}(\mu_{post}, h^{-1}_{post})$$

$$\text{where } \mu_{post} = h^{-1}_{post}(h_{prior}\mu_{prior} + n_k h_{k,d}\bar{y}_{k,d})$$

$$h_{post} = h_{prior} + n_k h_{k,d}$$

$$p(h_{k,d} \mid \boldsymbol{\mu}_k, \mathbf{z}, \mathbf{Y}) = \text{Gamma}(a_{post}, b_{post})$$

$$\text{where } a_{post} = a_{prior} + \frac{n_k}{2}$$

$$b_{post} = b_{prior} + \frac{1}{2}\sum_{n:z_n=k}(y_{n,d} - \mu_{k,d})^2$$

$$p(\boldsymbol{\pi} \mid \mathbf{z}, \mathbf{Y}) = \text{Dir}(\alpha_{post,1}, \ldots, \alpha_{post,K})$$

$$\text{where } \alpha_{post,k} = \alpha_{prior,k} + n_k$$

Each of these can be sampled from with readily available methods (e.g., `numpy.random.dirichlet()`)

Conditional for $\mathbf{z}$

We have already worked out that

$$p(z_n = k \mid \boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{Y}) = \text{Categorical}(q_{n1}, \ldots, q_{nK})$$
$$\text{where } q_{nk} = \frac{\pi_k \mathcal{N}(\mathbf{y}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'=1}^{K} \pi_k \mathcal{N}(\mathbf{y}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}$$

Conditioned on the cluster parameters, all z_n are independent. Sampling from this distribution is just a matter of enumerating

the probabilities, or using a library function for categorical distributions (e.g., `numpy.random.choice()`)

Gibbs Sampling More Generally

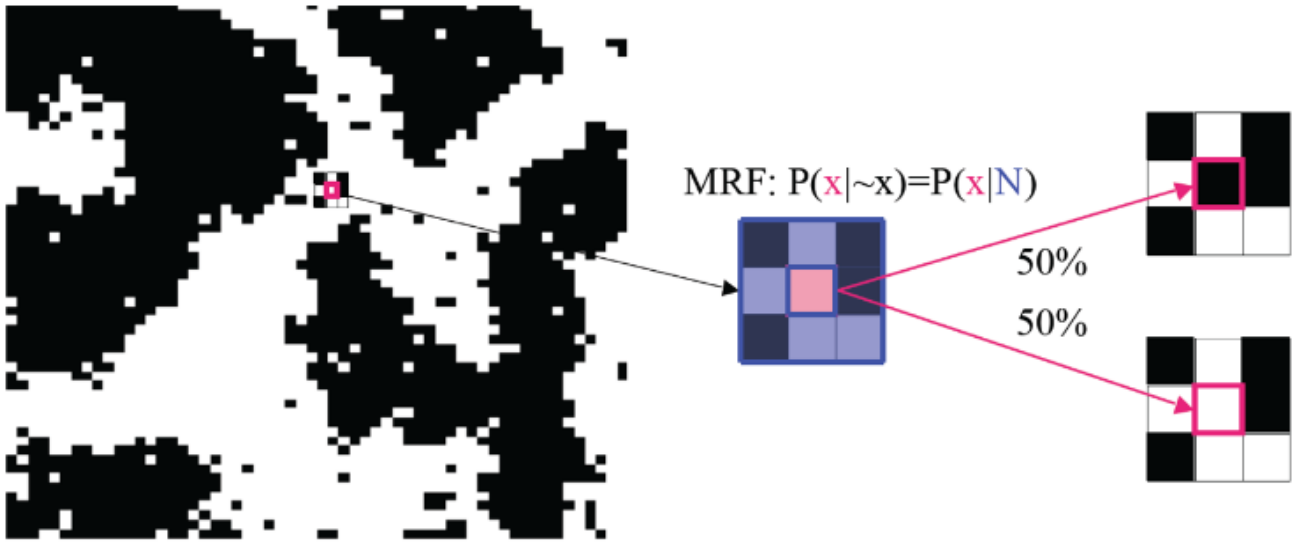
- Gibbs is a very general algorithm that can be used to sample from complicated probability distributions
- In general suppose we have $p(\boldsymbol{\theta}) = p(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M)$ where each $\boldsymbol{\theta}_m$ is a collection of random variables. We can generate samples from this distribution by
  1. Initializing each $\boldsymbol{\theta}_m$
  2. For $s = 1, \ldots, S$; For $m = 1, \ldots, M$:

     Sample $\boldsymbol{\theta}_m^{(s)}$ from $p(\boldsymbol{\theta}_m \mid \boldsymbol{\theta}_1^{(s)}, \ldots, \boldsymbol{\theta}_{m-1}^{(s)}, \boldsymbol{\theta}_{m+1}^{(s-1)}, \ldots, \boldsymbol{\theta}_M^{(s-1)})$

- Note that we can partition $\boldsymbol{\theta}$ into the $\boldsymbol{\theta}_m$ however we like, as long as certain conditions are met (more on this later)

Application: Image De-Noising

Consider a generative model of noisy images in which $\boldsymbol{\theta}$ represents the pixels of the "pure" image, and $\mathbf{Y}$ is the observed "corrupted" image.



(From Dellaert and Zhu tutorial)

Application: Image De-Noising

- We can define a prior on $\boldsymbol{\theta}$ that encourages neighboring pixels to be similar, and a likelihood on $\mathbf{Y}$ that encourages observed pixels to be close to the corresponding one in the pure image.
- Then, we Gibbs sample each pixel θ_m conditioned on the image and the other θs

$$p(\theta_m \mid \boldsymbol{\theta}_{-m}, \mathbf{Y}) \propto p(\theta_m \mid \boldsymbol{\theta}_{m-})p(y_m \mid \boldsymbol{\theta}_m)$$

Results from Two Priors

Weak Affinity to Neighbors Strong Affinity to Neighbors

(From Dellaert and Zhu tutorial)

Outline

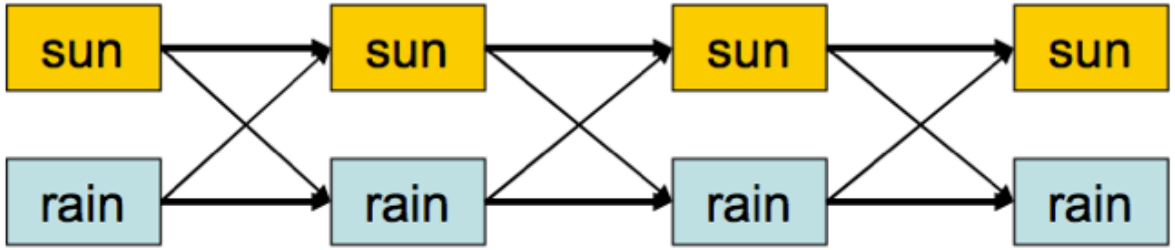
Gibbs Sampling in Mixture Models
- Sampling as an Approximation Method
- Gibbs Sampling Intuition and Motivation
- Concretely: The Algorithm

A Bit of MCMC Theory

Markov Chain Monte Carlo

- MCMC gets its name from the fact that it is a method for generating random samples (the "Monte Carlo" part) and the fact that each sample depends on exactly one previous sample (the "Markov Chain") part
- In general, involves sampling from $q(\boldsymbol{\theta}^{(s)} \mid \boldsymbol{\theta}^{(s-1)})$ for a suitable "transition kernel" q.
- Under certain conditions, $\boldsymbol{\theta}^{(s}$ for large s are distributed according to $p(\boldsymbol{\theta})$, which is determined by q.
- Idea: construct a q so that p is the target distribution we want to get samples from.

Markov Chains

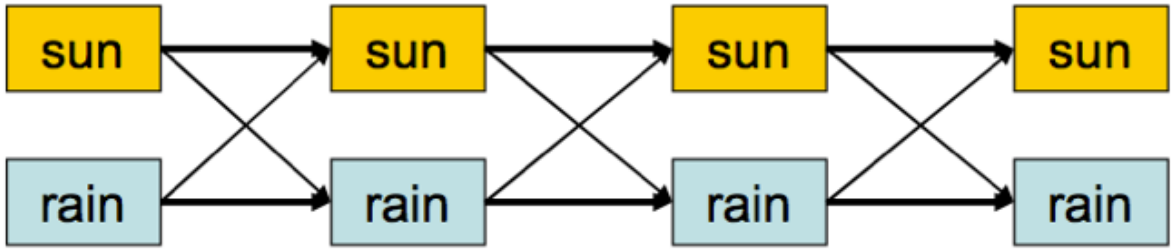


- Simple Markov chain: two possible states, $\{rain, sun\}$ at each time step t.
- Suppose $q(\cdot \mid \cdot)$ is defined by the transition matrix

$$\mathbf{Q} = \begin{pmatrix} & rain & sun \\ rain & 0.7 & 0.3 \\ sun & 0.1 & 0.9 \end{pmatrix}$$

where rows represent the "from" state and columns represent the "to" state. That is, $q(rain \mid rain) = 0.7$, $q(sun \mid rain) = 0.3$, etc.

Markov Chains



- Suppose $q(\cdot \mid \cdot)$ is defined by the transition matrix

$$\mathbf{Q} = \begin{pmatrix} & rain & sun \\ rain & 0.7 & 0.3 \\ sun & 0.1 & 0.9 \end{pmatrix}$$

- Question: If it is sunny today, what is the probability that it will be sunny tomorrow?
- If it is sunny today, what is the probability that it will be rainy two days from now?

Markov Chains

- Suppose $q(\cdot \mid \cdot)$ is defined by the transition matrix

$$\mathbf{Q} = \begin{pmatrix} & rain & sun \\ rain & 0.7 & 0.3 \\ sun & 0.1 & 0.9 \end{pmatrix}$$

- If it is sunny today, what is the probability that it will be rainy two days from now?
- We have

$$\begin{aligned} P(\theta_{t+2} = rain \mid \theta_t = sun) \\ &= \sum_s P(\theta_{t+1} = s, \theta_{t+2} = rain \mid \theta_t = sun) \\ &= \sum_s q(s \mid sun) q(rain \mid s) \\ &= (\mathbf{QQ})_{2,2} = (0.1, 0.9)^\top (0.7, 0.1) \\ &= 0.07 + 0.09 = 0.16 \end{aligned}$$

Markov Chains

- Suppose $q(\cdot \mid \cdot)$ is defined by the transition matrix

$$\mathbf{Q} = \begin{pmatrix} & rain & sun \\ rain & 0.7 & 0.3 \\ sun & 0.1 & 0.9 \end{pmatrix}$$

- If it is sunny today, what is the probability that it will be rainy a year from now?
- We have the recurrence relation

$$P(\theta_{t+m} \mid \theta_t) = \sum_s P(\theta_{t+m-1} = s, \theta_{t+m} \mid \theta_t)$$

- This is the matrix product of the $m-1$ step matrix with the one-step transition matrix.
- So the m-step transition matrix is just $\mathbf{Q}^m$

Simulation of $\mathbf{Q}^\infty$

- What does the "many step" transition matrix look like?
- Simulation
- We find that regardless of the starting state, there is a 1/4 chance of rain and a 3/4 chance of sun on any day far enough in the future.
- This is called the **stationary distribution** of the transition matrix $\mathbf{Q}$
- Mathematically, the row vector $\boldsymbol{\pi}$ is stationary w.r.t. $\mathbf{Q}$ if

$$\boldsymbol{\pi}\mathbf{Q} = \boldsymbol{\pi}$$

MCMC In General

- Gibbs sampling (and other MCMC algorithms) construct a transition kernel, $\mathbf{Q}$ (in Gibbs, subdivided into smaller transitions), so that the stationary distribution is the target distribution.
- Hence, after a sufficiently long time horizon, we get samples drawn from the posterior.

Beyond Gibbs: Metropolis-Hastings

Gibbs turns out to be a special case of the **Metropolis-Hastings** algorithm

The Metropolis-Hastings Algorithm: Want samples from $p(\boldsymbol{\theta})$

1. Initialize $\boldsymbol{\theta}^{(0)}$
2. For $s = 1, \ldots, S$
   (a) Draw a sample $\tilde{\boldsymbol{\theta}}^{(t)}$ from some **proposal density**, $\tilde{q}(\boldsymbol{\theta}^{(s)} \mid \boldsymbol{\theta}^{(s-1)})$
   (b) Compute the **acceptance probability**

$$A = \min\left\{1, \frac{p(\boldsymbol{\theta}^{(t)})\tilde{q}(\boldsymbol{\theta}^{(t-1)} \mid \boldsymbol{\theta}^{(t)})}{p(\boldsymbol{\theta}^{(t-1)})\tilde{q}(\boldsymbol{\theta}^{(t)} \mid \boldsymbol{\theta}^{(t-1)})}\right\}$$

   (c) Sample $u \sim \text{Unif}(0, 1)$. If $u \leq A$, set $\boldsymbol{\theta}^{(t)} = \tilde{\boldsymbol{\theta}}^{(t)}$. Otherwise, *reject* and set $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)}$.

Acceptance Probability Intuition

The acceptance probability is

$$A = \min\left\{1, \frac{p(\boldsymbol{\theta}^{(t)})\tilde{q}(\boldsymbol{\theta}^{(t-1)} \mid \boldsymbol{\theta}^{(t)})}{p(\boldsymbol{\theta}^{(t-1)})\tilde{q}(\boldsymbol{\theta}^{(t)} \mid \boldsymbol{\theta}^{(t-1)})}\right\}$$

Qualitative intuition: Tend to reject values that $\tilde{q}$ "oversuggests" and tend to stay at values that are "good" and hard to get back to later.

Another Special Case: Metropolis Algorithm

Proposal density: Gaussian random walk with fixed covariance $\boldsymbol{\Sigma}$

$$\tilde{q}(\boldsymbol{\theta} \mid \boldsymbol{\theta}') = \mathcal{N}(\boldsymbol{\theta} \mid \boldsymbol{\theta}', \boldsymbol{\Sigma})$$

Note that

$$\tilde{q}(\boldsymbol{\theta} \mid \boldsymbol{\theta}') = \tilde{q}(\boldsymbol{\theta}' \mid \boldsymbol{\theta})$$

Then we accept with probability

$$A = \min\left\{1, \frac{p(\boldsymbol{\theta}^{(t)})q(\boldsymbol{\theta}^{(t-1)} \mid \boldsymbol{\theta}^{(t)})}{p(\boldsymbol{\theta}^{(t-1)})q(\boldsymbol{\theta}^{(t)} \mid \boldsymbol{\theta}^{(t-1)})}\right\} = \min\left\{1, \frac{p(\boldsymbol{\theta}^{(t)})}{p(\boldsymbol{\theta}^{(t-1)}}\right\}$$

In other words, accept all "up-hill" steps, allow some "down-hill" steps.

Gibbs Yields Zero Rejections

In the mth part of a Gibbs iteration, we sample $\boldsymbol{\theta}'_m$ according to

$$\tilde{q}(\boldsymbol{\theta}'_m \mid \boldsymbol{\theta}_{-m}) = \frac{p(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{m-1}, \boldsymbol{\theta}'_m, \boldsymbol{\theta}_{m+1}, \ldots, \boldsymbol{\theta}_M)}{p(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{m-1}, \boldsymbol{\theta}_{m+1}, \ldots, \boldsymbol{\theta}_M)}$$

The ratio of this and its reverse is just $p(\boldsymbol{\theta}')/p(\boldsymbol{\theta})$, so the acceptance ratio becomes

$$\begin{aligned} A &= \min\left\{1, \frac{p(\boldsymbol{\theta}')q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')}{p(\boldsymbol{\theta})q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})}\right\} \\ &= \min\left\{1, \frac{p(\boldsymbol{\theta}')}{p(\boldsymbol{\theta})}\frac{p(\boldsymbol{\theta})}{p(\boldsymbol{\theta}')}\right\} = 1 \end{aligned}$$

When will MCMC (i.e., q) converge (to p)?

Some (rough) conditions (not rigorous![1]):

1. **Irreducibility**: Must be able to get anywhere from anywhere; no inescapable state vortexes (attractors) or isolated components of state space
2. **Aperiodicity**: No deterministic cycles

A sufficient (not necessary) condition is **detailed balance**:

$$p(\boldsymbol{\theta})q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}) = p(\boldsymbol{\theta}')q(\boldsymbol{\theta} \mid \boldsymbol{\theta}') \qquad \text{for all } \boldsymbol{\theta}, \boldsymbol{\theta}'$$

where $p(\boldsymbol{\theta})$ is a stationary distribution.

I.e., the expected number of jumps between any two states is the same in both directions.

[1] In fact, strictly, even these are only sufficient when state space is *finite*. See Neal (2003) (click for a link) for a good review.