Approx. Inference via Markov Chain Monte Carlo

Piyush Rai

Topics in Probabilistic Modeling and Inference (CS698X)

March 27, 2018

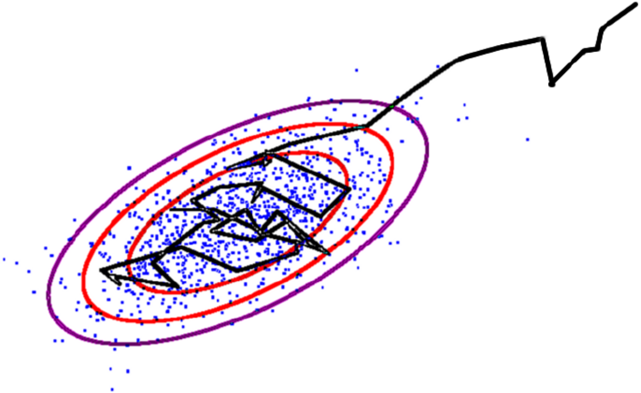
Markov Chain Monte Carlo (MCMC)

- Goal: Generate samples from some target distribution $p(\boldsymbol{z}) = \frac{\tilde{p}(\boldsymbol{z})}{Z}$, where $\boldsymbol{z}$ is high-dimensional
- Assume we can evaluate $p(\boldsymbol{z})$ at least up to a proportionality constant (i.e., can compute $\tilde{p}(\boldsymbol{z})$)
- Basic idea: MCMC uses a Markov Chain which, when converged, starts giving samples from $p(\boldsymbol{z})$

$$\underbrace{\boldsymbol{z}^{(1)} \rightarrow \boldsymbol{z}^{(2)} \rightarrow \boldsymbol{z}^{(3)} \rightarrow}_{\text{initial samples typically garbage}} \ldots \rightarrow \underbrace{\boldsymbol{z}^{(L-2)} \rightarrow \boldsymbol{z}^{(L-1)} \rightarrow \boldsymbol{z}^{(L)}}_{\text{after convergence, actual samples from } p(\boldsymbol{z})}$$

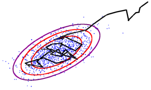
- Given a current sample $\boldsymbol{z}^{(\ell)}$ from the chain, MCMC generates the next sample $\boldsymbol{z}^{(\ell+1)}$ as
  - Use a proposal distribution $q(\boldsymbol{z}|\boldsymbol{z}^{(\ell)})$ to generate a candidate sample $\boldsymbol{z}^*$
  - Accept/reject $\boldsymbol{z}^*$ as the next sample based on an acceptance criterion (will see later)
  - If accepted, $\boldsymbol{z}^{(\ell+1)} = \boldsymbol{z}^*$. If rejected, $\boldsymbol{z}^{(\ell+1)} = \boldsymbol{z}^{(\ell)}$
- Note that in MCMC, the proposal distribution $q(\boldsymbol{z}|\boldsymbol{z}^{(\ell)})$ depends on the previous sample (unlike methods such as rejection sampling)

Markov Chain Monte Carlo (MCMC)



MCMC: The Basic Scheme

- MCMC chain run infinitely long (i.e., post-convergence) will give ONE sample from the target $p(\boldsymbol{z})$



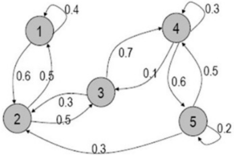
- But we usually require several samples to approximate $p(\boldsymbol{z})$. How do we get those?
  - Start at an initial $\boldsymbol{z}^{(0)}$. Using a prop. dist. $q(\boldsymbol{z}^{(\ell+1)}|\boldsymbol{z}^{(\ell)})$, run the chain long enough, say T_1 steps
  - Discard the first $(T_1 - 1)$ samples (called "burn-in" samples) and take the last sample $\boldsymbol{z}^{(T_1)}$
  - Continue from $\boldsymbol{z}^{(T_1)}$ up to T_2 steps, discard intermediate samples, take the last sample $\boldsymbol{z}^{(T_2)}$
    - This helps ensure that $\boldsymbol{z}^{(T_1)}$ and $\boldsymbol{z}^{(T_2)}$ are uncorrelated
  - Repeat the same for a total of S times
  - In the end, we have S i.i.d. samples from $p(\boldsymbol{z})$, i.e., $\boldsymbol{z}^{(T_1)}, \boldsymbol{z}^{(T_2)}, \ldots, \boldsymbol{z}^{(T_S)} \sim p(\boldsymbol{z})$
  - Note: Good choices for T_1 and $T_i - T_{i-1}$ are usually based on heuristics
  - Note: MCMC is an approximate method because we don't usually know what T_1 is "long enough"

MCMC: Some Basic Theory

- A first order Markov Chain assumes $p(\boldsymbol{z}^{(\ell+1)}|\boldsymbol{z}^{(1)},\ldots,\boldsymbol{z}^{(\ell)}) = p(\boldsymbol{z}^{(\ell+1)}|\boldsymbol{z}^{(\ell)})$
- A 1st order Markov Chain $\boldsymbol{z}^{(0)},\boldsymbol{z}^{(1)},\ldots,\boldsymbol{z}^{(L)}$ is a sequence of r.v.'s and is defined by the following
    - An initial state distribution $p(z^{(0)})$
    - A Transition Function (TF): $T_\ell(\boldsymbol{z}^{(\ell)} \rightarrow \boldsymbol{z}^{(\ell+1)}) = p(\boldsymbol{z}^{(\ell+1)}|\boldsymbol{z}^{(\ell)})$.
- TF defines a distribution over the values of next state given the value of the current state
- Assuming a discrete state-space, the TF is defined by a $K \times K$ probability table

Transition probabilities can be defined using a *K*x*K* table if **z** is a discrete r.v. with *K* possible values

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0.4 & 0.6 & 0.0 & 0.0 & 0.0 \\ 2 & 0.5 & 0.0 & 0.5 & 0.0 & 0.0 \\ 3 & 0.0 & 0.3 & 0.0 & 0.7 & 0.0 \\ 4 & 0.0 & 0.0 & 0.1 & 0.3 & 0.6 \\ 5 & 0.0 & 0.3 & 0.0 & 0.5 & 0.2 \end{array}$$

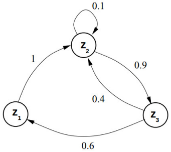


- Homogeneous Markov Chain: The TF is the same for all ℓ, i.e., $T_\ell = T$

MCMC: Some Basic Theory

- Consider the following simple TF

$$T = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0.1 & 0.9 \\ 0.6 & 0.4 & 0 \end{bmatrix}$$



- Consider the initial state distribution $p(\boldsymbol{z}^{(0)}) = [0.5, 0.2, 0.3]$
- Easy to see that $p(\boldsymbol{z}^{(0)}) \times T = [0.2, 0.6, 0.2] \Rightarrow$ distribution of $\boldsymbol{z}^{(1)}$
- Also easy to see that, after a few (say m) iterations, $p(\boldsymbol{z}^{(0)}) \times T^m = [0.2, 0.4, 0.4] = p(\boldsymbol{z})$ (say)
- For the above T, any choice of $p(\boldsymbol{z}^{(0)})$ leads to $p(\boldsymbol{z})$
  - Such a $p(\boldsymbol{z})$ is called the stationary/invariant distribution of this Markov Chain
- A Markov Chain has a stationary distribution if T has the following properties
  - Irreducibility: T's graph is connected (ensures reachability from anywhere to anywhere)
  - Aperiodicity: T's graph has no cycles (ensures that the chain isn't trapped in cycles)

MCMC: Some Basic Theory

- A Markov Chain has a stationary distribution $p(\boldsymbol{z})$ is the chain satisfies detailed balance

$$p(\boldsymbol{z})T(\boldsymbol{z} \to \boldsymbol{z}') = p(\boldsymbol{z}')T(\boldsymbol{z}' \to \boldsymbol{z})$$

- Integrating out (or summing over) both sides w.r.t. $\boldsymbol{z}'$ gives

$$p(\boldsymbol{z}) = \int p(\boldsymbol{z}')T(\boldsymbol{z}' \to \boldsymbol{z})d\boldsymbol{z}'$$

- Therefore $p(\boldsymbol{z})$ is a stationary distribution of this chain
- Thus a Markov Chain with detailed balance will always converge to a stationary distribution

Some MCMC Algorithms

Metropolis-Hastings (MH) Sampling (Hastings, 1970)

- Suppose we wish to generate samples from a distribution $p(\boldsymbol{z}) = \frac{\tilde{p}(\boldsymbol{z})}{Z_p}$
- Assume a proposal distribution $q(\boldsymbol{z}|\boldsymbol{z}^{(\tau)})$, e.g., $\mathcal{N}(\boldsymbol{z}|\boldsymbol{z}^{(\tau)}, \sigma^2\mathbf{I}_D)$
- In each step, draw $\boldsymbol{z}^* \sim q(\boldsymbol{z}|\boldsymbol{z}^{(\tau)})$ and accept the sample $\boldsymbol{z}^*$ with probability

$$A(\boldsymbol{z}^*, \boldsymbol{z}^{(\tau)}) = \min\left(1, \frac{\tilde{p}(\boldsymbol{z}^*)q(\boldsymbol{z}^{(\tau)}|\boldsymbol{z}^*)}{\tilde{p}(\boldsymbol{z}^{(\tau)})q(\boldsymbol{z}^*|\boldsymbol{z}^{(\tau)})}\right)$$

- The acceptance probability makes intuitive sense. Note the kind of $\boldsymbol{z}^*$ would it favor/unfavor:
  - It favors accepting $\boldsymbol{z}^*$ if $\tilde{p}(\boldsymbol{z}^*)$ has a higher value than $\tilde{p}(\boldsymbol{z}^{(\tau)})$
  - Unfavors $\boldsymbol{z}^*$ if the proposal distribution q unduly favors it (i.e., if $q(\boldsymbol{z}^*|\boldsymbol{z}^{(\tau)})$ is large)
  - Favors $\boldsymbol{z}^*$ if we can "reverse" to $\boldsymbol{z}^{(\tau)}$ from $\boldsymbol{z}^*$ (i.e., if $q(\boldsymbol{z}^{(\tau)}|\boldsymbol{z}^*)$ is large). Needed for good "mixing"
- Transition function of this Markov Chain: $T(\boldsymbol{z}^{(\tau)} \rightarrow \boldsymbol{z}^*) = A(\boldsymbol{z}^*, \boldsymbol{z}^{(\tau)})q(\boldsymbol{z}^*|\boldsymbol{z}^{(\tau)})$
- Exercise: Show that $T(\boldsymbol{z} \rightarrow \boldsymbol{z}^{(\tau)})$ satisfies the detailed balance property

$$T(\boldsymbol{z} \rightarrow \boldsymbol{z}^{(\tau)})p(\boldsymbol{z}) = T(\boldsymbol{z}^{(\tau)} \rightarrow \boldsymbol{z})p(\boldsymbol{z}^{(\tau)})$$

The MH Sampling Algorithm

- Initialize $\boldsymbol{z}^{(0)}$ randomly
- For $\ell = 0, \ldots, L-1$
  - Sample $u \sim \text{Unif}(0,1)$
  - Sample $\boldsymbol{z}^* \sim q(\boldsymbol{z}^*|\boldsymbol{z}^{(\ell)})$
  - If $u < A(\boldsymbol{z}^*, \boldsymbol{z}^{(\ell)}) = \min\left(1, \frac{\tilde{p}(\boldsymbol{z}^*)q(\boldsymbol{z}^{(\ell)}|\boldsymbol{z}^*)}{\tilde{p}(\boldsymbol{z}^{(\ell)})q(\boldsymbol{z}^*|\boldsymbol{z}^{(\ell)})}\right)$

$$\boldsymbol{z}^{(\ell+1)} = \boldsymbol{z}^* \qquad \text{(meaning: accepting with probability } A(\boldsymbol{z}^*, \boldsymbol{z}^{(\ell)}))$$

    else

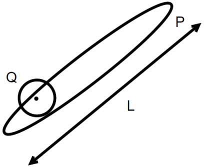
$$\boldsymbol{z}^{(\ell+1)} = \boldsymbol{z}^{(\ell)}$$

MH Sampling: Some Comments

- If proposal distrib. is symmetric, we get Metropolis Sampling algorithm (Metropolis, 1953) with

$$A(\boldsymbol{z}^*, \boldsymbol{z}^{(\tau)}) = \min\left(1, \frac{\tilde{p}(\boldsymbol{z}^*)}{\tilde{p}(\boldsymbol{z}^{(\tau)})}\right)$$

- Some limitations of MH sampling
  - MH can have a very slow convergence. Figure below: P is the target dist., Q is the proposal



Generic proposals use
$Q(x'; x) = \mathcal{N}(x, \sigma^2)$

σ **large** $\rightarrow$ **many rejections**

σ **small** $\rightarrow$ **slow diffusion:**
$\sim (L/\sigma)^2$ iterations required

  - Computing acceptance probability can be expensive: When $p(\boldsymbol{z}) = \frac{\tilde{p}(\boldsymbol{z})}{Z_p}$ represents a posterior distribution of some model, $\tilde{p}$ is the unnormalized posterior that depends on all the data (note: a lot of recent work on speeding up this step using subsets of data)

Gibbs Sampling (Geman & Geman, 1984)

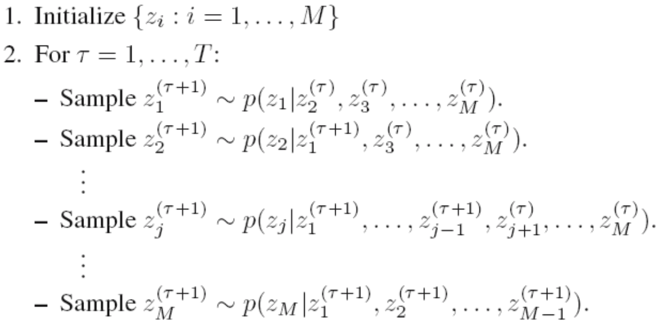
- Suppose we wish to sample from a joint distribution $p(\boldsymbol{z})$ where $\boldsymbol{z} = (z_1, z_2, \ldots, z_M)$
- However, suppose we can't sample from $p(\boldsymbol{z})$ but can sample from each conditional $p(z_i|\boldsymbol{z}_{-i})$
  - Can we done easily if we have a locally conjugate model
- Gibbs sampling uses the conditionals $p(z_i|\boldsymbol{z}_{-i})$ as the proposal distribution
- Gibbs sampling samples from these conditionals in a cyclic order
- Gibbs sampling is equivalent to Metropolis Hastings sampling with acceptance prob. $= 1$

$$A(\boldsymbol{z}^*, \boldsymbol{z}) = \frac{p(\boldsymbol{z}^*)q(\boldsymbol{z}|\boldsymbol{z}^*)}{p(\boldsymbol{z})q(\boldsymbol{z}^*|\boldsymbol{z})} = \frac{p(z_i^*|\boldsymbol{z}_{-i}^*)p(\boldsymbol{z}_{-i}^*)p(z_i|\boldsymbol{z}_{-i}^*)}{p(z_i|\boldsymbol{z}_{-i})p(\boldsymbol{z}_{-i})p(z_i^*|\boldsymbol{z}_{-i})} = 1$$

where we use the fact that $\boldsymbol{z}_{-i}^* = \boldsymbol{z}_{-i}$

Gibbs Sampling: Sketch of the Algorithm

M: Total number of variables, *T*: number of Gibbs sampling steps



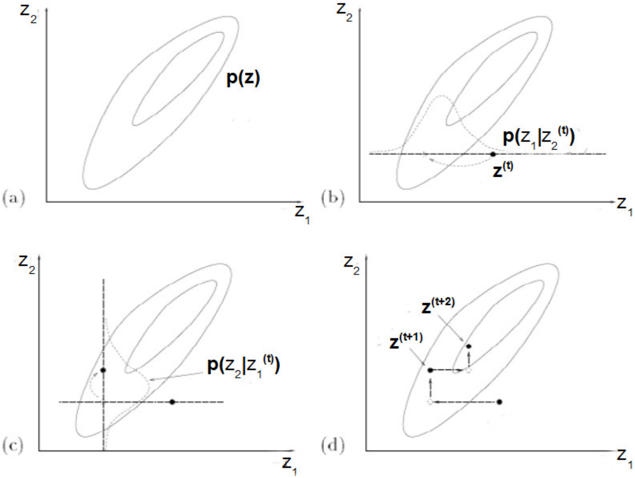
```
1. Initialize $\{z_i : i = 1, \ldots, M\}$
2. For $\tau = 1, \ldots, T$:
    – Sample $z_1^{(\tau+1)} \sim p(z_1 | z_2^{(\tau)}, z_3^{(\tau)}, \ldots, z_M^{(\tau)})$.
    – Sample $z_2^{(\tau+1)} \sim p(z_2 | z_1^{(\tau+1)}, z_3^{(\tau)}, \ldots, z_M^{(\tau)})$.
      ⋮
    – Sample $z_j^{(\tau+1)} \sim p(z_j | z_1^{(\tau+1)}, \ldots, z_{j-1}^{(\tau+1)}, z_{j+1}^{(\tau)}, \ldots, z_M^{(\tau)})$.
      ⋮
    – Sample $z_M^{(\tau+1)} \sim p(z_M | z_1^{(\tau+1)}, z_2^{(\tau+1)}, \ldots, z_{M-1}^{(\tau+1)})$.
```

Note: When sampling each variable from its conditional posterior, we use the most recent values of all other variables (this is akin to a co-ordinate ascent like procedure)

Note: Order of updating the variables *usually* doesn't matter (but see "Scan Order in Gibbs Sampling: Models in Which it Matters and Bounds on How Much" from NIPS 2016)

Gibbs Sampling: A Simple Example

Can sample from a 2-D Gaussian using 1-D Gaussians (recall that if the joint distribution is a 2-D Gaussian, conditionals will simply be 1-D Gaussians)



Gibbs Sampling: Some Comments

- One of the most popular MCMC algorithm
- Very easy to derive and implement for locally conjugate models
- Many variations exist, e.g.,
  - Blocked Gibbs: sample multiple variables jointly (sometimes possible)
  - Rao-Blackwellized Gibbs: Can collapse (i.e., integrate out) the unneeded variables while sampling. Also called "collapsed" Gibbs sampling
  - MH within Gibbs
- Instead of sampling from the conditionals, an alternative is to use the mode of the conditional.
  - Called the "Itearative Conditional Mode" (ICM) algorithm (doesn't give the posterior though)

Using MCMC Samples: An Example

- Consider making predictions in the PMF model: $\mathbf{R} = \mathbf{U}\mathbf{V}^\top + \boldsymbol{\epsilon}$, with $\mathbf{R}$ being $N \times M$, $\mathbf{U}$ being $N \times K$, $\mathbf{V}$ being $M \times K$, and each entry $r_{ij} = \boldsymbol{u}_i^\top \boldsymbol{v}_j + \epsilon_{ij}$
- Suppose our MCMC sampler (e.g., Gibbs sampler) gave us S samples $\{\mathbf{U}^{(s)}, \mathbf{V}^{(s)}\}_{s=1}^S$
- How to make predictions for a missing r_{ij} (its predictive mean, its predictive variance)?
- Given the samples $\{\mathbf{U}^{(s)}, \mathbf{V}^{(s)}\}_{s=1}^S$ from the posterior, we can approximate the mean of r_{ij} as

$$\mathbb{E}[r_{ij}|\mathbf{R}] = \mathbb{E}[\boldsymbol{u}_i^\top \boldsymbol{v}_j + \epsilon_{ij}] = \mathbb{E}[\boldsymbol{u}_i^\top \boldsymbol{v}_j] \approx \frac{1}{S}\sum_{s=1}^{S} \boldsymbol{u}_i^{(s)\top} \boldsymbol{v}_j^{(s)}$$

- The variance can be likewise approximated as

$$\begin{aligned}
\text{Var}[r_{ij}|\mathbf{R}] = \text{Var}[\boldsymbol{u}_i^\top \boldsymbol{v}_j + \epsilon_{ij}] &= \text{Var}[\boldsymbol{u}_i^\top \boldsymbol{v}_j] + \text{Var}[\epsilon_{ij}] \\
&= \mathbb{E}[(\boldsymbol{u}_i^\top \boldsymbol{v}_j)^2] - \left[\mathbb{E}[\boldsymbol{u}_i^\top \boldsymbol{v}_j]\right]^2 + \beta^{-1} \\
&\approx \frac{1}{S}\sum_{s=1}^{S}\left(\boldsymbol{u}_i^{(s)\top} \boldsymbol{v}_j^{(s)}\right)^2 - \left(\frac{1}{S}\sum_{s=1}^{S} \boldsymbol{u}_i^{(s)\top} \boldsymbol{v}_j^{(s)}\right)^2 + \beta^{-1}
\end{aligned}$$

- Question: Can't we average all samples to get a single $\mathbf{U}$ and a single $\mathbf{V}$ and use those?
  - No. Reason: Possible mode or label switching

Mode/Label Switching

- The posterior of most latent variable models has multiple modes
  - .. even if it is unimodal when the latent variables are known
- Each sampling iteration can give samples of latent variables from one of the modes
- Example: Consider a clustering model like GMM. Likelihood is invariant to label permutations
  - What one sample considers as cluster 1 may become cluster 2 for next sample
- Therefore averaging latent variables or parameters across samples can be meaningless
- Quantities not affected by permutations of latent variables can be safely averaged
  - Probability that two points belong to the same cluster (e.g., in GMM)
  - Predicting the mean/variance of a missing entry r_{ij} in matrix factorization

MCMC/Sampling vs Variational Inference

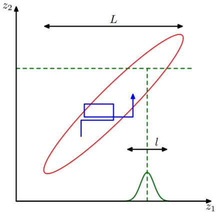
- MCMC gives samples from the true posterior when run infinitely long
    - But in practice we can never run the chain infinitely long
- VB only gives a locally optimal approximation of the posterior
    - But VB guarantees that we will give it in finite time!
- MCMC is quite general; doesn't assume simplifying assumptions such as mean-field
- Many VB methods (e.g.. BBVI) use sampling as a routine
- MCMC approximation is expensive storage-wise (need to store all the samples)
    - Also, computing any quantity that depends on the posterior is expensive with MCMC because we need to average using all the stored samples!

MCMC and Random Walk

- MCMC methods use a proposal distribution to draw the next sample given the previous sample

$$\theta^{(t)} \sim \mathcal{N}(\theta^{(t-1)}, \sigma^2)$$

- .. and then we accept/reject (if doing MH) or always accept (if doing Gibbs sampling)
- Such proposal distributions typically lead to a random-walk behavior (e.g., a zig-zag trajectory in Gibbs sampling) and may lead to very slow convergence (pic below: $\theta = [z_1, z_2]$)



- Can be especially critical when the components of θ are highly correlated
- Using the posterior's gradient may be beneficial for faster convergence (next class)