STA 602 - Intro to Bayesian Statistics

Lecture 18

Li Ma

Duke University

More general MCMC algorithms than Gibbs

- So far we have mostly been using Gibbs sampler to draw from posteriors where we don't know the normalizing constant in Bayes theorem (i.e., the marginal likelihood).
- Gibbs samplers require simple forms for all of the full conditionals that can be easily sampled from.
- This often is achievable for parameters in a Bayesian model (think about a DAG) where the upstream and downstream conditional distributions are specified in a conjugate way.
- What happens if not all full conditionals are available in such form?

More general MCMC algorithms than Gibbs

- Let's look at a more general MCMC sampler that *in principle* can work for *any* prior and likelihood specification.
  - Here "in principle' ' is important. Just because you can construct such a sampler doesn't mean it will be effective—the mixing and convergence can be very poor, and the more so when the number of parameters to be sampled grow.
- Most commonly, people will use a hybrid of Gibbs sampler and the more general sampler—drawing from the full conditionals directly when they are available, and use the more general samplers for updating the other parameters.

The Metropolis algorithm

- Recall that for the sampler chain to be a Markov chain, the parameter value in each iteration can only depend the value in the last iteration.
- The key to designing a new MCMC sampler is to choose the appropriate transition probabilities that generate the next value given the current value of a parameter.
- The most simple and well-known general-purpose MCMC sampler is the “Metropolis algorithm”.
- It borrows ideas from rejection sampling to design the Markov transition probabilities.

The Metropolis algorithm

- Let $\boldsymbol{\theta}$ be our parameter, and our goal is to generate samples of $\boldsymbol{\theta}$ from a target distribution $p(\boldsymbol{\theta})$. Examples of the target include the posterior $p(\boldsymbol{\theta}|\mathbf{x})$.
- Suppose after t iterations, its value is $\boldsymbol{\theta}^{(t)}$.
- To generate a value in the $(t+1)$st iteration, we do the following
  - Draw a new value $\boldsymbol{\theta}^*$ from a *proposal distribution* $q(\cdot|\boldsymbol{\theta}^{(t)})$ *that depends on the current value* $\boldsymbol{\theta}^{(t)}$ and is *symmetric*, that is $q(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2) = q(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1)$:

$$\boldsymbol{\theta}^* \sim q(\cdot|\boldsymbol{\theta}^{(t)}).$$

  - Then accept this draw as $\boldsymbol{\theta}^{(t+1)}$ with probability $r = \min\{1, p(\boldsymbol{\theta}^*)/p(\boldsymbol{\theta}^{(t)})\}$, and reject with probability $1-r$, in which case we inherit the current value of $\boldsymbol{\theta}^{(t)}$ as the new value $\boldsymbol{\theta}^{(t+1)}$.

The Metropolis algorithm

- The random rejection can be implemented by first drawing

$$u \sim \text{Unif}(0,1)$$

and setting

$$\boldsymbol{\theta}^{(t+1)} = \begin{cases} \boldsymbol{\theta}^* & \text{if } u < \frac{p(\boldsymbol{\theta}^*)}{p(\boldsymbol{\theta}^{(t)})} \\ \boldsymbol{\theta}^{(t)} & \text{otherwise.} \end{cases}$$

- What is the transition probability from a value $\boldsymbol{\theta}_1$ to a value $\boldsymbol{\theta}_2$?
- It is

$$p(\boldsymbol{\theta}_1 \to \boldsymbol{\theta}_2) = q(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1) \cdot \min\{1, p(\boldsymbol{\theta}_2)/p(\boldsymbol{\theta}_1)\}.$$

Convergence of the Metropolis chain

- Markov Chain theory guarantees that the chain so constructed

$$\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots$$

  will eventually converge to its stationary distribution p.

- One can easily check that the Markov chain satisfies the so-called *detailed-balance* condition

$$p(\boldsymbol{\theta}_1)p(\boldsymbol{\theta}_1 \rightarrow \boldsymbol{\theta}_2) = p(\boldsymbol{\theta}_2)p(\boldsymbol{\theta}_2 \rightarrow \boldsymbol{\theta}_1),$$

  which is a sufficient condition to guarantee convergence to p.

- The intuition is that the chain will eventually stay in an equilibrium with the flow between any two values balanced under the marginal distribution p.

Convergence of the Metropolis chain

- Specifically, we can check the detailed balance condition for the chain from the Metropolis algorithm:

$$
\begin{aligned}
& p(\boldsymbol{\theta}_1)p(\boldsymbol{\theta}_1 \to \boldsymbol{\theta}_2) \\
=& p(\boldsymbol{\theta}_1)q(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1)\min\{1, p(\boldsymbol{\theta}_2)/p(\boldsymbol{\theta}_1)\} \\
=& \min\{p(\boldsymbol{\theta}_1)q(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1), p(\boldsymbol{\theta}_2)q(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1)\} \\
=& \min\{p(\boldsymbol{\theta}_1)q(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2), p(\boldsymbol{\theta}_2)q(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2)\} \quad \text{(by symmetry of } q\text{)} \\
=& p(\boldsymbol{\theta}_2)q(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2)\min\{1, p(\boldsymbol{\theta}_1)/p(\boldsymbol{\theta}_2)\} \\
=& p(\boldsymbol{\theta}_2)p(\boldsymbol{\theta}_2 \to \boldsymbol{\theta}_1).
\end{aligned}
$$

With asymmetric proposals

- The requirement that the proposal q must be symmetric can be restrictive.
- A more general version of the algorithm removes the symmetry constraint on $q(\cdot|\cdot)$.
- Based on the above proof of detailed balance, we can modify the acceptance probability to

$$r = \min\left\{1, \frac{p(\boldsymbol{\theta}_2)q(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2)}{p(\boldsymbol{\theta}_1)q(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1)}\right\}$$

  then we can check detailed balance.
- Intuition: If our proposal "favors" $\boldsymbol{\theta}_2$ compared to $\boldsymbol{\theta}_1$, then the standard Metropolis acceptance will lead to too many draws of $\boldsymbol{\theta}_2$. To adjust for the oversampling due to the proposal, we must reduce the chance of accepting such a $\boldsymbol{\theta}_2$.
- This generalization is called the *Metropolis-Hastings* (MH) algorithm.
- The Metropolis algorithm is a special case when q is symmetric.

Convergence of MCMC chain from the MH algorithm

- Again, we can check the detailed balance condition

$$\begin{aligned}
&p(\boldsymbol{\theta}_1)p(\boldsymbol{\theta}_1 \to \boldsymbol{\theta}_2)\\
=&p(\boldsymbol{\theta}_1)q(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1) \cdot \min\left\{1, \frac{p(\boldsymbol{\theta}_2)q(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2)}{p(\boldsymbol{\theta}_1)q(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1)}\right\}\\
=&\min\{p(\boldsymbol{\theta}_1)q(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1), p(\boldsymbol{\theta}_2)q(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2)\}\\
=&p(\boldsymbol{\theta}_2)q(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2) \cdot \min\left\{1, \frac{p(\boldsymbol{\theta}_1)q(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1)}{p(\boldsymbol{\theta}_2)q(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2)}\right\}\\
=&p(\boldsymbol{\theta}_2)p(\boldsymbol{\theta}_2 \to \boldsymbol{\theta}_1).
\end{aligned}$$

- So the chain will converge to the stationary distribution p.

Gibbs sampler as an MH algorithm

- In fact, the very general MH algorithm contains Gibbs samplers as a special case as well.
- For a Gibbs sampler, the proposal q for a new value $\boldsymbol{\theta}^*$ is given by

$$q(\boldsymbol{\theta}^*|\boldsymbol{\theta}) = \begin{cases} p(\theta_i^*|\boldsymbol{\theta}_{-i}) & \text{if } \boldsymbol{\theta}_j^* = \boldsymbol{\theta} \text{ for all } j \neq i \\ 0 & \text{otherwise.} \end{cases}$$

- In this case, for any proposed value $\boldsymbol{\theta}^*$, which must differ from $\boldsymbol{\theta}$ by at most one margin i,

$$\begin{aligned} p(\boldsymbol{\theta})q(\boldsymbol{\theta}^*|\boldsymbol{\theta}) &= p(\boldsymbol{\theta})p(\theta_i^*|\boldsymbol{\theta}_{-i}) \\ &= p(\theta_i|\boldsymbol{\theta}_{-i})p(\boldsymbol{\theta}_{-i})p(\theta_i^*|\boldsymbol{\theta}_{-i}) \\ &= p(\boldsymbol{\theta}^*)p(\theta_i|\boldsymbol{\theta}_{-i}^*) = p(\boldsymbol{\theta}^*)q(\boldsymbol{\theta}|\boldsymbol{\theta}^*). \end{aligned}$$

- Therefore the acceptance ratio is

$$r = \min\left\{1, \frac{p(\boldsymbol{\theta}^*)q(\boldsymbol{\theta}^*|\boldsymbol{\theta})}{p(\boldsymbol{\theta})q(\boldsymbol{\theta}^*|\boldsymbol{\theta})}\right\} = 1$$

so the proposal is always accepted.

Random walk proposals

- A commonly used proposal distribution $q(\cdot|\boldsymbol{\theta})$ is to propose a random move $v \sim g(\cdot)$ from a distribution $g(\cdot)$ around $\mathbf{0}$ (symmetric for Metropolis but not necessarily so for MH) that doesn't depend on $\boldsymbol{\theta}$, and let $\boldsymbol{\theta}^* = \theta + v$.
- This is called a random walk proposal and g is the kernel for the random moves.
- Often $g(\cdot)$ is specified in terms of some parameters that control the "step size' ', δ.
  - For example, one may choose

$$g(\cdot|\delta) = \mathrm{N}(\mathbf{0}, \delta^2 \cdot I)$$

    or

$$g(\cdot|\delta) = \mathrm{Uniform}([-\delta, \delta]^d).$$

    which are both symmetric kernels.

Autocorrelation, mixing, and convergence

- The choice of the size of the move v as determined by the parameter δ determines the autocorrelation, mixing, and covergence of the MCMC.
- If δ is very large, the proposal tends to generate large moves, and if $p(\boldsymbol{\theta})$ is already quite high, it is likelihood to propses values of $p(\boldsymbol{\theta}^*) \ll p(\boldsymbol{\theta})$ leading to very small acceptance rate r.
  - High autocorrelation, poor mixing, and slow convergence.
- If δ is very small, the proposal tends to generate tiny moves, and the chain though have high acceptance rate, moves very slowly in the parameter space.
  - High autocorrelation, poor mixing, and slow convergence.
- Both are not desirable and will lead to huge Monte Carlo error.

Choice of move size

- Ideally, one should choose a δ that strikes a balance between accepting the proposals and the size of the moves.
  - A rule-of-thumb is to set δ so that the acceptance rate is about 20-50%, larger the dimensionality of $\boldsymbol{\theta}$ the smaller one would expect, and closer to 50% for 1-dimensional θ.
  - Generally, I don't recommend applying MH on too many dimensions (> 5) if possible, as it is very hard to choose a good proposal.

Combining MH and Gibbs

- Since Gibbs is a special case of MH, mixing Gibbs and MH still gives a special case of MH and will lead to a valid MCMC algorithm.
- Generally, let $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_d)$, and we want to sample from the joint distribution $p(\boldsymbol{\theta})$.
- After initialization, we again can proceed as we did for Gibbs sampling
- For $t = 1, 2, \ldots$
  - For $i = 1, \ldots, d$,
    - Update the value of $\theta_i^{(t)}$ based on the full conditional θ_i given the current values of the other parameters $\boldsymbol{\theta}_{-i}^{curr}$.

Combining MH and Gibbs

- More specifically,
  - If the full conditional $p(\theta_i|\boldsymbol{\theta}_{-i})$ can be sampled from directly, do so (as in standard Gibbs).
  - If the full conditional $p(\theta_i|\boldsymbol{\theta}_{-i})$ cannot be directly sampled from, then do an MH step
    - First proposal a value from $q_i(\cdot|\boldsymbol{\theta}_{-i}^{curr})$ which can now depend on the current value
      $$\theta_i^* \sim q_i(\theta_i|\boldsymbol{\theta}_{-i}^{curr}).$$
    - Accept the proposal and set $\theta_i^{(t)} = \theta_i^*$ with probability
      $$r = \min\left\{1, \frac{p(\theta_i^*|\boldsymbol{\theta}_{-i}^{curr})q_i(\theta_i^{(t-1)}|\boldsymbol{\theta}_{-i}^{curr})}{p(\theta_i^{(t-1)}|\boldsymbol{\theta}_{-i}^{curr})q_i(\theta_i^*|\boldsymbol{\theta}_{-i}^{curr})}\right\}$$
    - If not accept, set $\theta_i^{(t)} = \theta_i^{(t-1)}$.