10-708: Probabilistic Graphical Models, Spring 2015

17 : Markov Chain Monte Carlo

Lecturer: Eric P. Xing *Scribes: Heran Lin, Bin Deng, Yun Huang*

1 Review of Monte Carlo Methods

1.1 Overview

Monte Carlo methods can be used to approximate expectations by averaging over samples. Suppose x is a random variable with density $p(x)$. Then for a function f,

$$\mathbb{E}[f(x)] = \int f(x)p(x)dx \approx \frac{1}{J}\sum_{j=1}^{J} f(x^{(j)})$$

where $x^{(j)} \sim p(x)$ are i.i.d samples drawn from $p(x)$. Note that the variance of the sum is

$$\mathsf{Var}\left[\frac{1}{J}\sum_{j=1}^{J} f(x^{(j)})\right] = \frac{1}{J^2}\cdot J \cdot \mathsf{Var}[f(x^{(j)})] = O\left(\frac{1}{J}\right)$$

which decreases as J gets larger. So the approximation will be more accurate as we obtain more samples.

Here is an example of using Monte Carlo methods to integrate away weights in Bayesian neural networks. Let $y(x) = f(x, w)$ for response y and input x, and let $p(w)$ be the prior over the weights w. The posterior distribution of w given the data $\mathcal{D}$ is $p(w|\mathcal{D}) \propto p(\mathcal{D}|w)p(w)$ where $p(\mathcal{D}|w)$ is the likelihood. For a test input x_*, we approximate the distribution of the response variable y_* as

$$p(y_*|x_*, \mathcal{D}) = \int p(y_*|w, x_*)p(w|\mathcal{D})dw \approx \frac{1}{J}\sum_{j=1}^{J} p(y_*|w^{(j)}, x_*)$$

where $w^{(j)} \sim p(w|\mathcal{D})$. If we know the posterior $p(w|\mathcal{D})$ up to a normalizing constant, then rejection sampling or importance sampling can be used to obtain samples from the posterior.

However, one must be cautious when attempting to sample from prior distributions. Consider the following approximation to the marginal distribution using samples $w^{(j)}$ drawn from the prior distribution $p(w)$.

$$p(\mathcal{D}) = \int p(\mathcal{D}|w)p(w)dw \approx \frac{1}{J}\sum_{j=1}^{J} p(\mathcal{D}|w^{(j)}).$$

The above sampling scheme is "dangerous" in that we may get very different answers in multiple runs. In other words, the variance of the above finite approximation is very high. This is because most samples from $p(w)$ will have low likelihood $p(\mathcal{D}|w)$, so the sum is dominated by a small fraction of terms which may differ greatly across runs.

1.2 A Few Monte Carlo Sampler

Sampling from a density function $p(x)$ is equivalent to drawing samples uniformly from the area under $p(x)$ (and keeps only the x values). Here we briefly describe a few samplers, some of which are topics of the last lecture.

Inverse Transform Sampling As a starting point, we consider a basic Monte Carlo sampler using a uniform distribution and variable transformation. Let $x \sim U(0,1)$ and $y = f(x)$. Then the density of y is

$$p(y) = p(x)\frac{dx}{dy} = \frac{dx}{dy}.$$

So

$$x = g(y) = \int_{-\infty}^{y} p(y')dy'$$

where $g(y)$ is the CDF of y and we have $y = g^{-1}(x)$. Therefore, if we sample x from the uniform distribution $U(0,1)$ and transform the sample using the inverse CDF of y, then we will obtain a sample from the density $p(y)$.

Reject Sampling In rejection sampling, we sample x_0 from a proposal distribution $q(x)$ such that $kq(x) \geq \tilde{p}(x)$ where k is some constant and $\tilde{p}(x)$ is the unnormalized version of $p(x)$. Then we sample u_0 from the uniform distribution $U(0, kq(x_0))$. In this way, (x_0, u_0) is drawn uniformly under the curve $kq(x)$. We accept the sample x_0 if $u_0 \leq \tilde{p}(x_0)$.

Importance Sampling The idea of importance sampling is illustrated by the following approximation.

$$\int f(x)p(x)dx = \int f(x)\frac{p(x)}{q(x)}q(x)dx \approx \frac{1}{J}\sum_{j=1}^{J}\frac{p(x^{(j)})}{q(x^{(j)})}f(x^{(j)}), \quad x^{(j)} \sim q(x).$$

Ancestral Sampling To sample a Bayesian network, we can perform ancestral sampling. We start by sampling top level variables from their marginal distributions and sample other node conditioned on the samples of their parent nodes. For example, suppose we want to sample from the following distribution.

$$p(A,B,C,D,E) = P(A)P(B)P(C|A,B)P(D|B,C)P(E|C,D).$$

We proceed as follows.

$$A \sim P(A), \quad B \sim P(B), \quad C \sim P(C|A,B), \quad D \sim P(D|B,C), \quad E \sim P(E|C,D).$$

However, ancestral sampling may not work if we cannot decompose the distribution into low dimensional conditional distributions. For example, in undirected graphical models it is often hard to compute the partition function. Another example can be a conditional distribution $P(A,B,C,D|E)$ from the above Bayesian network, where we may not know the marginal distribution $P(E)$.

1.3 Limitations for Monte Carlo Methods

Rejection sampling and importance sampling may not work well in high dimensions. Here is an example. Let $p(x)$ and $q(x)$ be the density function of $N(0,I)$ and $N(0,\sigma^2 I)$, respectively. For the rejection sampling

to work, we must have $\sigma \geq 1$. Then the acceptance rate of rejection sampling is

$$\int \frac{p(x)}{kq(x)} q(x)dx = \frac{1}{k}$$

where we must have $k = \sigma^D$ (recall the density at origin for normal distributions with mean zero). So the acceptance rate will be very low if the dimensionality D is large. For importance sampling, one can show that the variance of the importance weight is $\left(\frac{\sigma^2}{2-1/\sigma^2}\right)^{D/2} - 1$, which also becomes large as D increases.

In general, for $kq(x) \geq p(x)$ to hold, the ratio of the volume of $p(x)$ to the volume outside $p(x)$ tends to zero as D increases, so it is very inefficient to use the proposal distribution to sample from $p(x)$.

2 Markov Chain Monte Carlo (MCMC)

Markov Chain Monte Carlo (MCMC) methods can be used to draw samples from high dimensional distributions without knowing much about the distribution. In MCMC, sample z_{i+1} is drawn from a transition probability $T(z_{i+1}|z_i)$ where z_i is the previous sample. So the samples $z_1, z_2, \ldots$ form a Markov chain. The transition probability depends on an adaptive proposal density $q(z_{i+1}|z_i)$ and an acceptance rule. We can also write the transition probability as $T(z_{i+1} \leftarrow z_i)$ for clarity. The adaptive proposal density has the capability of moving the sampler to multiple regions of high density in $p(x)$, even if we start with a bad proposal which puts most mass around only one mode of $p(x)$.

2.1 Example: the Metropolis Algorithm

The Metropolis algorithm is a special case of the Metropolis-Hastings algorithm to be described later. We first sample x' from a symmetric adaptive proposal distribution conditioned on the previous sample (e.g. sampling from a Gaussian distribution with mean x and variance σ^2). We accept the new sample with probability $\min(1, p(x')/p(x))$. If the new sample x' is rejected, we duplicate the previous sample x as our next sample. This is different from rejection sampling where we discard rejected samples.

2.2 Properties of Markov Chains

As the samples in MCMC form a Markov chain, we must ensure that it converges to the original distribution $p(x)$ we want to sample. To address the problem, we first investigate a few properties of Markov chains. We say a Markov chain is *homogeneous* if the transition probability (or *transition operator*) $T_i(z_{i+1} \leftarrow z_i)$ is the same for all steps i. A distribution $p^*(x)$ is *invariant* with respect to a Markov chain if

$$p^*(z) = \sum_{z'} T(z \leftarrow z')p^*(z').$$

$p^*(x)$ is said to satisfy *detailed balance* if

$$p^*(z')T(z \leftarrow z') = p^*(z)T(z' \leftarrow z)$$

which means that it is equally probable to start from state z' and move to z or vice versa. Detailed balance is a sufficient but not necessary condition for invariance. Suppose a distribution $p^*(x)$ satisfies detailed balance. Then

$$\sum_{z'} T(z \leftarrow z')p^*(z') = \sum_{z'} p^*(z)T(z' \leftarrow z) = p^*(z) \sum_{z'} T(z' \leftarrow z) = p^*(z).$$

So detailed balance implies invariance.

We define a *reverse operator* as follows.

$$\tilde{T}(z \leftarrow z') = \frac{T(z' \leftarrow z)p^*(z)}{\sum_z T(z' \leftarrow z)p^*(z)}.$$

If a distribution $p^*(x)$ is invariant (or *stationary*) of the transition operator T, then

$$\tilde{T}(z \leftarrow z') = \frac{T(z' \leftarrow z)p^*(z)}{p^*(z')} \propto T(z' \leftarrow z)p^*(z).$$

Also, if an operator satisfies detailed balance, then its reverse operator is itself. We say the *generalized detailed balance* property holds if

$$T(z' \leftarrow z)p^*(z) = \tilde{T}(z \leftarrow z')p^*(z').$$

Generalized detailed balance is both the sufficient and necessary condition for invariance.

Finally, we state that we can use Markov chains to sample from distribution $p^*(x)$ if $p^*(x)$ is invariant with respect to the Markov chain and the *egodicity* property holds, meaning that $p(z_i) \to p^*(z)$ as the step $i \to \infty$ regardless of the initial distribution $p(z_0)$.

One problem remains. How do we create transition operators in the first place? One option is through a linear combination of a set of base transitions as

$$T(z \leftarrow z') = \sum_{k=1}^{K} \alpha_k B_k(z', z).$$

This is analogous to generating new kernels by combining existing ones. Alternatively, we can construct transition operators as

$$T(z \leftarrow z') = \sum_{z_1} \cdots \sum_{z_{K-1}} B_1(z', z_1) \dots B_K(z_{K-1}, z).$$

If a distribution is invariant with respect to all base transitions, then it is also invariant with respect to both transition operators constructed above. If all base transitions satisfies detailed balance, then the first transition operator also satisfies detailed balance. However, for the second transition operator constructed above to satisfy detailed balance, we need additionally require that the application of base transitions are symmetric, in the form of $B_1 \dots B_{K-1} B_K B_K B_{K-1} \dots B_1$.

2.3 Metropolis-Hastings Algorithm

The Metropolis algorithm is a special case of the Metropolis-Hastings algorithm where the proposal function is symmetric. Generally speaking, the Metropolis-Hastings algorithm designs a Markov process by constructing transition probabilities from the proposal density $q(x'|x)$ which can be any fixed density from which we can draw samples. Following is the algorithm.

1. Pick an initial state x at random.
2. Sample from the proposal distribution as $x \sim q(x'|x)$.
3. Accept with probability $\min\left(1, \frac{p(x')q(x|x')}{p(x)p(x'|x)}\right)$. (Note that we do not require that $p(x)$ be normalized since the normalizer cancels in the acceptance ratio.)

4. If rejected, the next state is a repeat of the current state (which is different from rejection sampling).

5. Iterate Step 2 to 4.

In order for the Metropolis-Hastings algorithm to work, two conditions are needed: (1) simulated sequence is a Markov Chain with a unique stationary distribution, (2) the stationary distribution needs to be equal to original true distribution. Condition (1) holds if the Markov chain is irreducible, aperiodic and not transient. The latter two conditions (aperiodic and not transient) hold for a random walk on any proper distribution, and irreducibility holds if the random walk has positive probability of eventually reaching any state from any other state. This is true for all sensible proposal densities we could use. Now we show why (2) holds by proving the detailed balance condition as follows.

$$\begin{aligned} p(x)T(x' \leftarrow x) &= p(x)q(x'|x)\min\left(1, \frac{p(x')q(x|x')}{p(x)q(x'|x)}\right) \\ &= \min(p(x)q(x'|x), p(x')q(x|x')) \\ &= p(x')q(x|x')\min\left(1, \frac{p(x)q(x'|x)}{p(x')q(x|x')}\right) \\ &= p(x')T(x \leftarrow x'). \end{aligned}$$

The choice of $q(x|x')$ should fulfill some other technical requirements. Generic proposals use $q(x'|x) = \mathcal{N}(x, \sigma^2)$ and require a suitable step size (σ): large σ results in too many rejections; small σ results in slow diffusion (poor exploration) (empirically 40% to 70% acceptance rate may indicate suitable step size). This is a drawback of the MH algorithm. In addition, it struggles badly with multi-modal distributions. However, the benefit of MH is that it is simple to implement and it is reasonable for sampling from correlated high dimensional distributions.

2.4 Gibbs Sampling

Gibbs sampling is a method for sampling from distributions over at least two dimensions. It can be viewed as a Metropolis-Hastings method in which a sequence of proposal distributions $q(x|x')$ are defined in terms of the conditional distributions of the joint distribution $p(x)$ (and the acceptance probabilities becomes one). It is assumed that, whilst $p(x)$ is too complex to draw samples from directly, its conditional distributions $p(x_i|x_{j\neq i})$ are tractable to work with. For many graphical models (but not all) these one-dimensional conditional distributions are straightforward to sample from. The algorithm is as follows.

1. Initialize x_1, x_2, …, x_K;

2. Sample conditional distribution: $x_1^{(t+1)} \sim p(x_1|x_2^{(t)}, x_3^{(t)}, \ldots, x_K^{(t)})$;

3. Sample conditional distribution: $x_2^{(t+1)} \sim p(x_2|x_1^{(t+1)}, x_3^{(t)}, \ldots, x_K^{(t)})$;

4. Sample conditional distribution: $x_3^{(t+1)} \sim p(x_3|x_1^{(t+1)}, x_2^{(t+1)}, \ldots, x_K^{(t)})$, etc.;

5. Iterates Step 2 to 4 for all variables.

In practice the single random variables to be updated are usually not chosen at random based on a distribution q but subsequently in fixed predefined order. The corresponding algorithm is often referred to as periodic Gibbs Sampler.

It is not hard to see that the original joint distribution is the stationary distribution of the Markov chain defined by these transition probabilities by showing that the detailed balance condition holds (Similar to the proof of Metropolis-Hastings algorithm, the properties of being irreducible, aperiodic and not transient can be satisfied.). Assuming at each transition step we pick a random variable x_i with a probability $q(i)$ given by a strictly positive probability distribution q, and $\mathbf{x}$ and $\mathbf{y}$ differ only in the state of exactly one variable x_i, i.e., $y_j = x_j$ for $j \neq i$ and $y_i \neq x_i$, then it holds:

$$\begin{aligned} p(\mathbf{x})T(\mathbf{y} \leftarrow \mathbf{x}) &= p(\mathbf{x})q(i)p(y_i|\mathbf{x}_{j\neq i}) \\ &= p(x_i, \mathbf{x}_{j\neq i})q(i)\frac{p(y_i, \mathbf{x}_{j\neq i})}{p(\mathbf{x}_{j\neq i})} \\ &= p(y_i, \mathbf{x}_{j\neq i})q(i)\frac{p(x_i, \mathbf{x}_{j\neq i})}{p(\mathbf{x}_{j\neq i})} \\ &= p(\mathbf{y})q(i)p(x_i|\mathbf{x}_{j\neq i}) \\ &= p(\mathbf{y})T(\mathbf{x} \leftarrow \mathbf{y}). \end{aligned}$$

The advantage of Gibbs sampling are as follows: (1) it is easy to evaluate the conditional distributions, (2) conditionals may be conjugate and we can sample from them exactly, (3) conditionals will be lower dimensional and we can apply rejection sampling or importance sampling. However, the major drawback is that when variables have strong dependencies it is hard to move around. We can introduce auxiliary variables to help move around when such high dimensional variables are correlated.

2.5 Assessing Convergence

Running MCMC can be a "black magic". However, we can have following strategies for assessing convergence. References can be checked in the course slides.

1. **Diagnostics:** Plot autocorrelations, compute Gelman-Rubin statistics.
2. **Using thinning, multiple runs, burn-in:** We can keep every K-th sample, or perform multiple runs, or discard a "burn-in" period.
3. **Unit tests:** We can run on small-scale versions of our problem, and reasonable inferences on synthetic data drawn from our model.

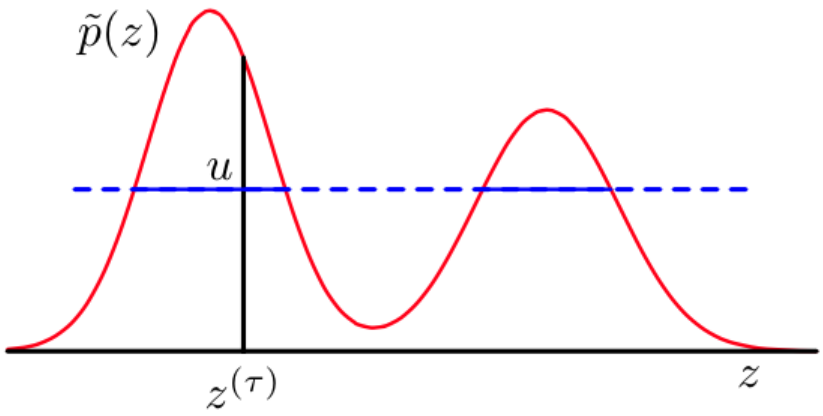
3 Slice Sampling

One disadvantage of the Metropolis algorithm is that it is sensitive to step size. Small step size may cause random walk; big step size may cause high rejection rate. Slice sampling will overcome this by automatically adjusting step size to fit the distribution. The algorithm still requires unnormalized distribution $\tilde{p}(x)$.

3.1 Auxiliary Variables

Auxiliary variables are random variables that do not exist in the model but are introduced into the model to facilitate sampling. Assume we sample from density $p(x)$, and the form of $p(x)$ may be hard for sampling. We can introduce an auxiliary variable μ. If the joint distribution $p(x, \mu)$ is easy to navigate, and the conditional distribution $p(x|\mu)$ and $p(\mu|x)$ have simple forms, then sampling from $p(x, \mu)$ with auxiliary variable μ will be much easier.

3.2 Algorithm



The slice sampling can be summarized as:

1. Initialize $z^{(\tau)}$
2. Sample $\mu \sim \text{Uniform}(0, \tilde{p}(z^{(\tau)}))$
3. Sample uniformly from the slice $\{z : \tilde{p}(z) > \mu\}$. Step out to find the foundries.

μ is the auxiliary variable. Given a value $z^{(\tau)}$, we can evaluate $\tilde{p}(z)$. μ can be uniformly sampled from 0 to $\tilde{p}(z)$. In the next step, fix μ and sample z uniformly from the slice through the distribution defined by $\{z : \tilde{p} > \mu\}$. The sample (z, μ) are uniformly distributed under the area of $\tilde{p}(z)$. We can obtain each sample z by dropping out μ (marginalization).

The set $\{z : \tilde{p}(z) > \mu\}$ can be computationally expensive or unfeasible. Bracket slice can be applied. We can put a horizontal bracket slice containing $(z^{(\tau)}, \mu)$ and test each end points of bracket to see if they are located within the slice, that is, $\tilde{p}(z) > \mu$. If either end does, extend this end until it is out of the slice region. Sample next z uniformly in the bracket, if new z is located within the slice ($\tilde{p}(z) > \mu$), keep it; otherwise shrink the bracket width until finding new z in the slice ($\tilde{p}(z) > \mu$). It is worth noting that the first sample $(z^{(\tau)}, \mu)$ should be kept in bracket during shrinking.

3.3 Advantages and Disadvantages

There are several advantages of slice sampling:

1. Very automatic: lack of tunable free parameters, proposal distribution, etc.
2. No rejections
3. Great choice when there is little knowledge about the distribution we are sampling from

There are also some disadvantages. When it goes to multidimensional distribution, one may sample each variable in turn using 1D slice sampling, in a manner similar to Gibbs sampling. However, this method severely suffer from complication introduced by dimensionality.

4 Hamiltonian Monte Carlo

Hamiltonian Monte Carlo or hybrid Monte Carlo is a Metropolis method that uses gradient information for the continuous space. Hamiltonian Monte Carlo helps us to avoid the basic random walk behavior from the previous simple Metropolis method.

4.1 Fundamentals

Generally a probability distribution can be written as

$$p(x) = \frac{1}{Z}\exp(-E(x))$$

where $E(x)$ is potential energy of system in state x. It is usually easy to evaluate $E(x)$ as well as gradient of $E(x)$. The gradient indicates which direction to go to find states of higher probability. It seems wasteful to ignore this information in the case of random walk Metropolis method.

In Hamiltonian Monte Carlo, $x = \{x_i\}$ is the state variable under continuous time denoted by τ. Define intermediate momentum variable ν:

$$\nu_i = \frac{dx_i}{d\tau}.$$

Define kinetic energy $K(\nu) = \nu^T\nu/2$, the total energy of system is sum of its potential and kinetic energy:

$$H(x, \nu) = E(x) + K(\nu).$$

The joint density which is used to create sample is

$$p(x, \nu) = \frac{1}{Z_H}\exp(-H(x, \nu)) = \frac{1}{Z_H}\exp(-E(x))\exp(-K(\nu)).$$

Since the density is separable, the marginal distribution of x is $p(x) = \frac{1}{Z}\exp(-E(x))$. If we can sample from $p(x, \nu)$, we can simply discard the samples of ν for samples of x.

In Hamiltonian dynamics, $\hat{x}$ and $\hat{\nu}$ can be updated via leapfrog discretization:

$$\hat{r_i}(\tau + \epsilon/2) = \hat{r_i}(\tau) - \frac{\epsilon}{2}\frac{\partial E}{\partial x_i}(\hat{x}(\tau))$$

$$\hat{x_i}(\tau + \epsilon) = \hat{x_i}(\tau) + \epsilon\hat{r_i}(\tau + \epsilon/2)$$

$$\hat{r_i}(\tau + \epsilon) = \hat{r_i}(\tau + \epsilon/2) - \frac{\epsilon}{2}\frac{\partial E}{\partial x_i}(\hat{x}(\tau + \epsilon))$$

4.2 Advantages and Disadvantages

Advantages:

1. Very efficient with good settings of τ and ϵ
2. State-of-the-art for sampling from posteriors over Bayesian neural networks

Disadvantages:

1. Very difficult to tune τ and ϵ
2. Hamiltonian helps with local exploration, but not with multimodality

5 Annealing Methods

5.1 Simulated Annealing

When working with MCMC, convergence rate of Markov chain and the decorrelation time between independent samples can be problematic. It may be less frequent to transit from a isolated high probability region to another high probability region. Simulated annealing is proposed to deal with this problem. The distribution is modified with temperature parameter T:

$$p(x) = \frac{1}{Z(T)} \exp\left(\frac{-E(x)}{T}\right).$$

The original distribution corresponds $T = 1$. We sample from this modified distribution. In simulated annealing, T value is initialized > 1 and gradually decreases to 1. This procedure deemphasizes transition between high probability region and low probability region. Therefore it increases chance of transition from a high probability region to another. Another form of simulated annealing is to decompose energy function $E(x)$:

$$p(x) = \frac{1}{Z(T)} \exp\left(-E_0(x) + \frac{E_1(x)}{T}\right)$$

where $E_0(x)$ may have some nice property such as separable and convex; $E_1(x)$ represents difference between $E_0(x)$ and true energy function $E(x)$.

5.2 Parallel Tempering

In parallel tempering, the original state space coexists with stationary states $\{p_k\}$:

$$p(x^{(k)}) = \prod_{k=1}^{K+1} p_k(x^{(k)}).$$

The marginal and stationary distributions form $K + 1$ levels. At each state of MCMC, adjacent levels can switch with each other. Since each state at each level is independent under target distribution, we can apply each independent transition operator T_k to each stationary distribution p_k.