CSC 446 Notes: Lecture 13

Typed by Shujie Chen

1 The Problem

We have already studied how to calculate the probability of a variable or variables using the message passing method. However, there are some times when the structure of the graph is too complicated to be calculated. The relation between the diseases and symptoms is a good example, where the variables are all mixed together and brings the graph a high tree width. Another case is that of continuous variable, where during the message passing,

$$r_{m\to n} = \int f(\vec{x}_m) \prod_{n'} q_{n'\to m}(x_{n'}) \mathrm{d}(\vec{x}_m \backslash x_n).$$

If this integration can not be calculated, what can we do to evaluate the probability of variables? This is what sampling is used for.

2 How to Sample a Continuous Variable: Basics

Now let us forget the above for a moment, say if we want to sample for a continuous variable, how can we ensure that the points we pick up satisfy the distribution of that variable? This question is easy for variables with uniform distribution, since we can generate random numbers directly using a computer. For some complicated distributions, we could use the inverse of cumulative distribution function (CDF) to map the uniform distribution onto the required distribution to generate samples, where the CDF for a distribution with probability distribution function (PDF) of P is

$$\mathrm{CDF}(x) = \int_{-\infty}^{x} P(t)\mathrm{d}t.$$

For example, if we want to sample from a variable with standard normal distribution, the points we pick up are calculated from

$$X = \mathrm{erf}^{-1}(x),$$

where x is drawn from a uniform distribution, and

$$\mathrm{erf}(x) = \int_{0}^{x} \mathcal{N}(t, 0, 1)\mathrm{d}t,$$

We could play the same trick for many other distributions. However, there are some distributions which do not have a closed-form integral to calculate their CDF, which makes the above method fail. Under such conditions, we could turn to a framework called *Markov chain Monte Carlo* (MCMC).

3 The Metropolis-Hastings Algorithm

Before discussing this method in more detail, let us review some basic properties of Markov chains. A first-order Markov chain is a series of random variables such that each variable depends only on its previous state, that is,

$$x^t \sim P(x^t | x^{t-1}).$$

Our goal is to find a Markov chain which has a distribution similar to a given distribution which we want to sample from, so that by running the Markov chain, we get results as if we were sampling from the original distribution. In other words, we want to have the Markov chain that eventually be able to 1) explore over the entire space of the original distribution, 2) reflect the original PDF.

The general algorithm for generating the samples is called *the Metropolis-Hastings algorithm.* Such an algorithm draws a candidate

$$x' \sim Q(x'; x^t),$$

and then accepts it with probability

$$\min\left\{1, \frac{P(x')Q(x^t; x')}{P(x^t)Q(x'; x^t)}\right\}.$$

The key here is the function Q, called *proposed distribution* which is used to reduce the complexity of the original distribution. Therefore, we have to select a Q that is easy to sample from, for instance, a Gaussian function. Note that there is a trade-off on choosing the variance of the Gaussian, which determines the step size of the Markov chain. If it is too small, it will take a long time, or even make it impossible for the states of the variable to go over the entire space. However, if the variance is too large, the probability of accepting the new candidate will become small, and thus it is possible that the variable will stay on the same state for ever. All these extremes will make the chain fail to simulate the original PDF.

If we sample from P directly, that is $Q(x'; x^t) = P(x')$, we have

$$\frac{P(x')Q(x^t; x')}{P(x^t)Q(x'; x^t)} = 1,$$

which means that the candidate we draw will always be accepted. This tells us that Q should approximate P.

By the way, how do we calculate $P(x)$? There are two cases.

- Although we cannot get the integration of $P(x)$, $P(x)$ itself is easy to compute.
- $P(x) = f(x)/Z$, where $Z = \int f(x)\mathrm{d}x$ is what we do not know. But since we know $f(x) = ZP(x)$, we could just substitute $f(x)$ instead of $P(x)$ in calculating the probability of acceptance of a candidate.

4 Proof of the method

In this section, our goal is to prove that the Markov chain generated by the Metropolis-Hastings algorithm has a unique stationary distribution. We will first introduce some basics about the definition of the stationary distribution, and the method to prove this "stationary". Then we will apply those knowledge to accomplish our goal.

(a) Stationary distribution

A distribution with respect to a Markov chain is said to be *stationary* if the distribution remains the same before and after taking one step in the chain, which could be denoted as

$$\Pi^t = T \times \Pi^{t-1} = \Pi^{t-1},$$

or

$$\Pi_i = \sum_j T_{ij}\Pi_j, \quad \forall i,$$

where Π is a vector which contains the stationary distribution of the state of the variable in each step with its element $\Pi_i = P(x = i)$, and T is the transition probability matrix where its element $T_{ij} = P(x^t = i|x^{t-1} = j)$ denotes the probability that the variable transits from state j to i. For example, the two Markov chains in item 1a and item 1b all have a stationary distribution $\Pi = \begin{bmatrix}0.5\\0.5\end{bmatrix}$.

The stationary distribution of a Markov chain could be calculated by solving the equation

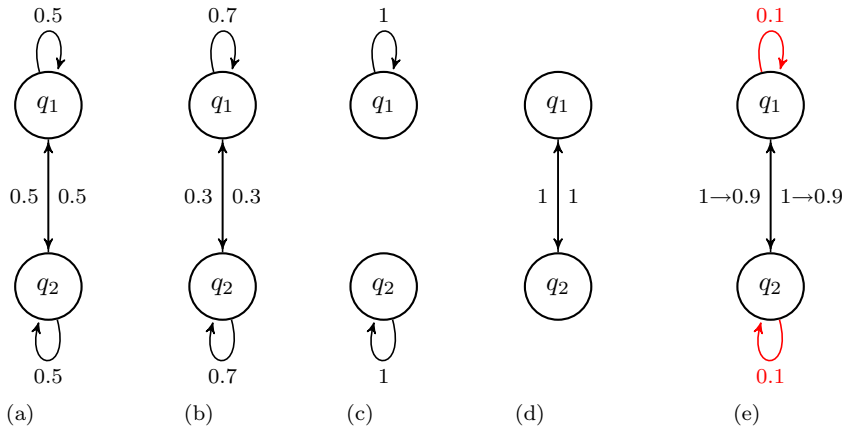

Figure 1: Example Markov Chains

$$\begin{cases} T\Pi = & \Pi \\ \sum_i \Pi_i = & 1 \end{cases}.$$

Note that there might be more than one stationary distribution for a Markov chain. A rather simple example would be a chain with a identity transition matrix shown in item 1c.

If a Markov chain has a stationary distribution and the stationary distribution is unique, it is ensured that it will converge eventually to that distribution no matter what the original state the chain is.

(b) Detailed Balance: Property to ensure the stationary distribution

Once we know a Markov chain is uniquely stationary, then we can use it to sample from a given distribution. Now, we will see a sufficient (but not necessary) condition for ensuring a Π is stationary, which is a property of the transition matrix called *Detailed Balance*. The definition of such a property is

$$\forall ij, \quad T_{ij}\Pi_j = T_{ji}\Pi_i,$$

which means $P_{i\to j} = P_{j\to i}$, and is also called reversibility due to the symmetry of the structure.

Starting from such definition, we have

$$\forall i, \quad \sum_j T_{ij}\Pi_j = \sum_j T_{ji}\Pi_i = \Pi_i \sum_j T_{ji}.$$

Note that $\sum_j T_{ji} = 1$, we come up with

$$\forall i, \quad \sum_j T_{ij}\Pi_j = \Pi_i \cdot 1 = \Pi_i,$$

which is exactly the second definition of stationary distribution we have just discussed. Therefore, if a distribution makes the transition matrix of a Markov chain satisfy detailed balance, that distribution is the stationary distribution of that chain. Note that although a periodic Markov chain like that shown in item 1d satisfies detailed balance, we do not call it stationary. This because it will not truly converge and thus is not guaranteed to approximate the original PDF. What is more, it is often the case that we add a probability like shown in item 1e to avoid such a periodic circumstance.

Note that the Detailed Balance does not ensure the uniqueness of the stationary distribution of a Markov chain. However, such uniqueness is necessary, or the Markov chain would not go to the PDF we want.

What we could do is that, when we construct the chain at the very beginning, we make the chain such that 1) any state is reachable for any other and 2) the chain is aperiodic. Under that condition, we could ensure the uniqueness of the stationary distribution.

(c) Final proof

Now, let us be back to the Metropolis-Hastings algorithm and prove that the transition matrix of its Markov chain has the detailed balance property. If we can prove that, it is obvious that such a Markov chain has a unique stationary distribution.

According to the Metropolis-Hastings algorithm, the transition probability of the Markov chain of the algorithm is

$$T(x';x) = Q(x';x) \cdot \min\left\{1, \frac{P(x')Q(x;x')}{P(x)Q(x';x)}\right\}$$

If $x' = x$, then it is automatically detailed balancing due to the symmetry of the definition of detailed balance. To be specific, the condition of detailed balance, which is

$$\forall ij, T_{ij}\Pi_j = T_{ji}\Pi_i,$$

will always be valid if $i = j$, which is just the case of $x' = x$.

For the circumstances that $x' \neq x$, by using the distributive property of multiplication, the transition probability is derived as,

$$T(x';x) = \min\left\{Q(x';x), \frac{P(x')Q(x;x')}{P(x)}\right\}.$$

Multiply both sides by $P(x)$, it turns out that

$$\begin{aligned} T(x';x)P(x) &= \underbrace{\min\left\{Q(x';x)P(x), P(x')Q(x;x')\right\}}_{\text{symmetric for } x \text{ \& } x'} \\ &= T(x;x')P(x') \end{aligned}$$

Therefore, we proved the detailed balance of the transition matrix, and thus the Markov chain of the Metropolis-Hastings algorithm does have a stationary distribution, which means that we could use such a Markov chain to simulate the original PDF.

5 Gibbs Sampling

Now, back to the very first problem of this class, we want to get the result of

$$P(x_k) = \sum_{\vec{x}\setminus x_k} \frac{1}{Z} \prod_m f(\vec{x}_m)$$

without knowing Z. We could use the Gibbs Sampling, shown in Algorithm 1, where $x_{\neg k}$ means all the

Algorithm 1: Gibbs Sampling

for $k = 1 \dots K$ **do**
  $x_k \sim P(x_k|x_{\neg k}) = \frac{1}{Z'} \prod_{m \in M(k)} f(\vec{x}_m)$;

variables x except x_k. Note that the Gibbs Sampling is actually a particular instance of the Metropolis-Hastings algorithm where the new candidate is always accepted. This is proved as follows.

Substitute

$$\begin{cases} P(x) = P(x_{\neg k})P(x_k|x_{\neg k}) \\ P(x') = P(x'_{\neg k})P(x'_k|x'_{\neg k}) \\ Q(x';x) = P(x'_k|x_{\neg k}) \\ Q(x;x') = P(x_k|x'_{\neg k}) \\ x'_{\neg k} = x_{\neg k} \end{cases}$$

into the probability of the accepting the new candidate, we have

$$\begin{aligned} \frac{P(x')Q(x;x')}{P(x)Q(x';x)} &= \frac{P(x'_{\neg k})P(x'_k|x'_{\neg k})P(x_k|x'_{\neg k})}{P(x_{\neg k})P(x_k|x_{\neg k})P(x'_k|x_{\neg k})} \\ &= 1. \end{aligned}$$

Therefore, the Gibbs Sampling algorithm will always accept the candidate.