10-708: Probabilistic Graphical Models 10-708, Spring 2014

18 : Advanced topics in MCMC

Lecturer: Eric P. Xing *Scribes: Jessica Chemali, Seungwhan Moon*

1 Gibbs Sampling (Continued from the last lecture)

1.1 Collapsed Gibbsed Sampling

Collapsed Gibbs Sampling is a variant of the basic Gibbs sampler, which is a popular inference algorithm for topic models. In topic models, Collapsed Gibbs Sampler integrates out (or marginalizes over) topic vectors π and topics B (Figure 1. Therefore, one only needs to sample word-topic assignments z, which greatly reduces the complexity for computing $P(z_i|z_{-i}, w)$. The algorithm for the sampling process can thus be simplified as drawing z_i^{t+1} from $P(z_i|z_{-i}, w)$ at each iteration for all variables $z = z_1, z_2, \cdots, z_n$.

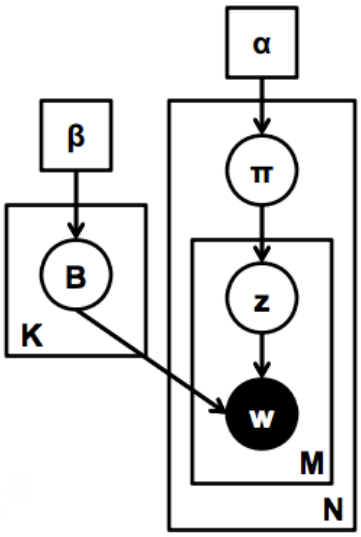

Figure 1: Topic Model Representation

Note that $P(z_i|z_{-i}, w)$ is essentially a product of two Dirichlet-Multinomial conditional distributions (thus "collapseing" Dirichlet distributions (priors)):

$$\begin{aligned} P(z_i|z_{-i}, w) &= P(w|z, \beta)P(z|\alpha) \\ &\propto \frac{n_{-i,j}^{(w_i)} + \beta}{n_{-i,j}^{(\cdot)} + W\beta} \cdot \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,j}^{(d_i)} + T\alpha} \end{aligned} \tag{1}$$

where $n_{-i,j}^{(w_i)}$ is the number of word positions (excluding w_i) such that $w_a = w_i$, $z_a = j$, and $n_{-i,j}^{(d_i)}$ is the

number of word positions in the current document d_i such that $z_a = j$. In other words, $P(w|z, \beta)$ is the word-topic term where B is integrated out, and the second term $P(z|\alpha)$ refers to the document-topic term where π is collapsed (integrated out). Note that both $P(w|z, \beta)$ and $P(z|\alpha)$ are Dirichlet distributions.

1.2 Acceptance of Gibbs Sampling

In this section, we present how Gibbs sampling is a special case of Metropolis-Hastings algorithm with a proposal acceptance rate of 1.

The Gibbs Sampling proposal distribution is given as:

$$Q(x_i', x_{-i}|x_i, x_{-i}) = P(x_i'|x_{-i}) \tag{2}$$

Applying this proposal to Metropolis-Hastings algorithm to compute acceptance yields:

$$\begin{aligned} A(x_i', x_{-i}|x_i, x_{-i}) &= min(1, \frac{P(x_i'|x_{-1})Q(x_i, x_{-i}|x_i', x_{-i})}{P(x_i|x_{-i})Q(x_i', x_{-i}|x_i, x_{-i})}) \\ &= min(1, \frac{P(x_i'|x_{-1})P(x_i|x_{-1})}{P(x_i|x_{-i})P(x_i'|x_{-1})}) \\ &= min(1, \frac{P(x_i'|x_{-1})P(x_i)P(x_i|x_{-1})}{P(x_i|x_{-i})P(x_i)P(x_i'|x_{-1})}) \\ &= min(1, 1) = 1 \end{aligned} \tag{3}$$

2 Slice Sampling

2.1 Auxiliary Variables

Before we begin our discussion of Slice Sampling, we define the auxiliary variables which many MCMC algorithms rely on. Auxiliary variables are extra random variables that are not from the original model, but introduced to allow us to sample model random variables in creative ways. Suppose that one wishes to sample from the density $\pi(x)$ using MCMC. An auxiliary varialbe u and its conditional distribution $\pi(x|u)$ can be introduced, giving the joint distribution $\pi(x, u) = \pi(x) \cdot \pi(u|x)$. If $\pi(x|u)$ and $\pi(u|x)$ have simple forms, the auxiliary variable u makes it much easier to navigate $\pi(x, u)$.

2.2 Auxiliary variable MCMC algorithm

Slice sampling is an auxiliary variable MCMC algorithm, where the key idea is to introduce an auxiliary variable a by uniformly sampling the area under $P'(x) = aP(x)$, instead of $P(x)$. The assumption is that $P'(x)$ is much cheaper than $P(x)$, thus avoiding the computational complexity involved with computing $P(x)$. For example, consider a Markov Random Field which is defined as $P(x) = \frac{1}{a}exp(bx)$, where a is the normalizer. Often times the normalizer a is intractable to evaluate, whereas the exponential evaluation is considered much more tractable. As such, we employ Slice Sampling which only requires a much easier evaluation of $P'(x) = exp(bx)$.

2.3 Algorithm

The Slice Sampling algorithm can be summarized as below:

Iterate between:

1. Sample h from $Q(h|x) = Uniform[0, P'(x)]$
2. Sample x from $Q(x|h)$ where $Q(x|h) = 1$ if $P'(x) \geq h$, 0 otherwise.

where h is an auxiliary variable. Note that the second step requires finding the set $\{x|P'(x) \geq h\}$, which can be computationally expensive e.g. when we cannot analytically find $\{x|P'(x) \geq h\}$ from complex distributions. One solution is to employ the "bracketing" strategy combined with the rejection sampling. First, we draw a random bracket width w, and place the bracket on (x_{old}, h). We then incremently increase the bracket width w until it reaches the endpoints a and b where $P'(a) < h$ and $P'(b) < h$. We then uniformly sample from within the bracket and reject samples x where $P'(x) < h$.

Note that at convergence, the samples (x, h) will be uniformly distributed under the area of $P'(x)$. Therefore, if we marginalize out h, we can get samples from $P(x) = (1/a)P'(x)$ without ever having to evaluate normalizer $(1/a)$. Given the set of samples $(x_1, h_1), (x_2, h_2), \cdots$, we can obtain X which is a set of samples from $P(x)$ simply by dropping h (marginalization).

3 Reversible Jump MCMC

Reversible Jump MCMC is a sampling method based on Metropolis-Hastings used for sampling from different model spaces (e.g. models with different numbers of parameters) [1]. This problem occurs in Bayesian model selection where one is interested in jointly inferring the best model and its parameters.

3.1 Definitions

Let x be the current model random variable (note that its dimensionality is unknown, and changes during the RJMCMC procedure), and let u be an auxiliary variable that is used by the algorithm to perform "dimension matching". Let m index the model that the algorithm is at, and $P(x|m)$ be the distribution over the data given model m.

Proposals

The RJMCMC algorithm uses 2 types of proposals:

- Model proposal $j(m'|m)$ switches from model m to model m'. The models need not have the same dimensionality, but the proposal must be reversible.
- Data proposal $q(x', u'|m \to m', x, u)$ proposes (x', u') under model m' starting from (x, u) under model m.

Mapping function

$h(m, m')$ is a deterministic bijective function from the space of model m to the space of model m', that computed how (x, u) under model m maps unto (x', u') under model m'.

3.2 RJMCMC Algorithm

1. Initialize x, u, m.
2. Repeat until convergence:
   (a) Propose a new model m' using the model proposal $j(m'|m)$
   (b) Propose a new model state (x',u') using the data proposal $q(x', u'|m- > m', x, u)$
   (c) Compute the acceptance probability:

$$A(m', x', u'|m, x, u) = min\big(1, \frac{P(x'|m')}{P(x|m)} \times \frac{j(m|m')}{j(m'|m)} \times \frac{q(x', u'|m \to m', x, u)}{q(x, u|m' \to m, x', u')} \times |det(\frac{\partial h_{m,m'}(x, u)}{\partial(x, u)})|\big)$$

3.3 Converging to the Stationary Distribution in RJMCMC

Recall that the goal of any MCMC algorithm is to allow sampling from a distribution of interest $\pi(x)$. The MCMC algorithm is therefore constructed in such a way that a stationary distribution is guaranteed to exist and that the distribution of interest is the stationary distribution of the Markov chain. In the MCMC algorithms we have previously introduced in class, the detailed balance condition was the vehicle to guarantee that the MCMC has such a distribution. Detailed balance consists of having:

$$\pi(x)T(x'|x) = \pi(x')T(x|x').$$

where x and x' are different states of the system and $T(.)$ is the transition kernel from one state to the other. Here:

$$T(x'|x) = j(m'|m) \times q(x', u'|m \to m', x, u) \times A(x', u'|x, u) \tag{4}$$

However in the problem that RJMCMC solves, x and x' have different dimensions, and we cannot write such a detailed balance equation since $P(x)$ and $P(x')$ are different distributions, that are defined on different spaces!

How does the RJMCMC guarantee the existence of a stationary distribution then? By making sure that a more fundamental property is satisfied:

$$\pi(x') = \int \pi(x)T(x'|x)dx \tag{5}$$

In words, the requirement is that if one integrates over all starting points x, the stationary distribution does not change under the transition kernel.

Now we can show how the RJMCMC acceptance rate is constructed. Let $g(.) = j(.) \times q(.)$. Then the requirement is that

$$\int P(x)g(x', u'|x, u)A(x', u'|x, u)dxdu = \int P(x')g(x, u|x', u')A(x, u|x', u')dx'du' \tag{6}$$

Note that we omit the model indicator m, because the dimensionality of m fully determines the model identity. Also note that the integration in the LHS occurs on a different space that the one in the RHS (This is the challenge we faced in using the detailed balance to construct the acceptance rate). However, here we perform a change of variable from (x',u') to (x,u) and get

$$\int P(x)g(x',u'|x,u)A(x',u'|x,u)dxdu = \int P(x')g(x,u|x',u')A(x,u|x',u')|det(\frac{\partial h_{(x',u'),(x,u)}(x,u)}{\partial(x,u)})|dxdu \tag{7}$$

Now both sides of the equation are integrated over the same space. This equation holds if for every (x,u) we have that

$$P(x)g(x',u'|x,u)A(x',u'|x,u) = P(x')g(x,u|x',u')A(x,u|x',u')|det(\frac{\partial h_{(x',u'),(x,u)}(x,u)}{\partial(x,u)})| \tag{8}$$

which leads to the acceptance rate:

$$A(m',x',u'|m,x,u) = min\big(1, \frac{P(x'|m')}{P(x|m)} \times \frac{j(m|m')}{j(m'|m)} \times \frac{q(x',u'|m \to m',x,u)}{q(x,u|m' \to m,x',u')} \times |det(\frac{\partial h_{m,m'}(x,u)}{\partial(x,u)})|\big) \tag{9}$$

Here $\frac{P(x'|m')}{P(x|m)}$ is the ratio of model probabilities and $\frac{j(m|m')}{j(m'|m)} \times \frac{q(x',u'|m \to m',x,u)}{q(x,u|m' \to m,x',u')} \times |det(\frac{\partial h_{m,m'}(x,u)}{\partial(x,u)})|$ is equivalent to the ratio of proposals in the Metropolis-Hastings algorithm. Note that unlike the previously introduced MCMC algorithms, $P(x'|m')$ and $P(x|m)$ are not the stationary distribution (they are actually different). This raises a question: **What does the RJMCMC converge to, or what is the stationary distribution of an RJMCMC algorithm?**

It converges into an **implicit** distribution over the space of all possible models' states. However it is not possible to know it it explicitly, since any x the algorithm is in will belongs to one specific model space.

3.4 Clustering Example

We now give a clustering example where RJMCMC is used to infer jointly the number of clusters and the assignments of the data to the clusters.

Let $m = 1, 2, 3, \cdots$ denote the number of clusters and let $P(x|m,c)$ be the probability of the observed data given the cluster centers c, under model m.

Let the model proposal $j(m'|m)$ be as follows:

- $m' = m - 1$ with probability $0.5 - p$ is used to decrease the number of clusters,
- $m' = m + 1$ with probability $0.5 - p$ is used to increase the number of clusters, and
- $m' = m$ with probability $2p$ is used to change the cluster centers.

Note that the reverse moves have the same probability as forward moves. The form of both the data proposal $q(x',c',u'|m \to m',x,c,u)$ and the mapping function $h_{m,m'}(x,c,u)$ depends on the m and m'.

Finally, due to the design of $h(.)$, the abs-det-Jacobian is always equal to 1! Let us consider the 3 cases separately:

1. $m' = m$
   (a) First pick a cluster center $c_i \in 1, 2, \cdots, m$ uniformly to change its assignment.

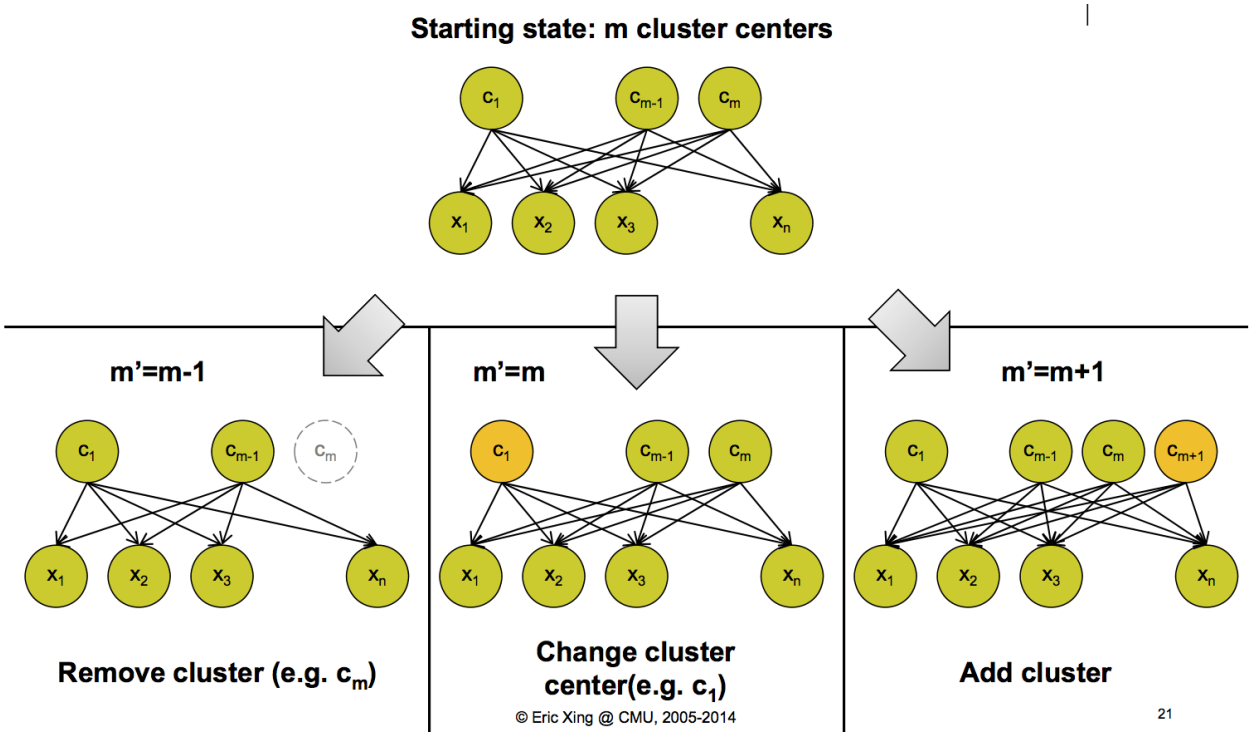

Figure 2: The 3 different cases of RJMCMC: $m' = m - 1$, $m' = m$ and $m' = m + 1$

(b) Next draw a new cluster center u from the proposal $q_{center}(u)$,

(c) Finally set $c'_i = u$.

Here $q(x', c', u'|m \to m', x, c, u) = \frac{1}{m} q_{center}(u)$ and $h_{m,m'}(x, c, u) = [c'_1, c'_2, \cdots, c'_m u']$ where $c'_j = c_j$ if $j \neq i$, $c'_i = u$, $c_i = u'$.

2. m' = m - 1

   Pick a cluster c_i uniformly at random to remove.

   Here $q(x', c', u'|m \to m', x, c, u) = \frac{1}{m}$ and $h_{m,m'}(x, c, u) = [c'_1, c'_2, \cdots, c'_{m-1} u']$ where $c'_j = c_j$ if $j < i$, $c'_j = c_{j+1}$ if $j > i$ $u = \emptyset$, $u' = c_i$.

3. m' = m+1

   Draw a cluster center u according to a proposal $q_{center}(u)$.

   Here $q(x', c', u'|m \to m', x, c, u) = q_{center}(u)$ and $h_{m,m'}(x, c, u) = [c'_1, c'_2, \cdots, c'_{m+1}]$ where $c'_j = c_j$ if $j \leq m$, $c'_{m+1} = u$, $u' = \emptyset$.

Important properties

- Note that all model changes are reversible:
  - We can get any cluster number m,
  - We can change the location of any cluster center c_i,

  This ensures that we converge to the stationary distribution.

- The abs-det-Jacobian is always 1: We designed h for this to be true, which makes the computation much faster.

References

[1] D. I. Hastie and P. J. Green, "Model choice using reversible jump markov chain monte carlo," *Statistica Neerlandica*, vol. 66, no. 3, pp. 309–338, 2012.