

Instructors: Piyush Rai  
Authors: Gurpreet Singh  
Date: October 18, 2017

## Approximate Inference and Sampling Methods

### 1 Locally Conjugate Models

Inference with multiple unknowns (parameters or hyperparameters) becomes tricky, and usually have intractable posteriors as well. Special methods such as EM or MLE-II have to be used.

We can approximate these posteriors using the ideo of **local** or **conditional conjugacy**. Essentially, the routine is to update one parameter keeping others fixed (similar to EM).

#### 1.1 Local Conjugacy

Often is the case that the overall posterior  $p(\Theta | \mathbf{X}) = \frac{p(\mathbf{x} | \Theta)p(\Theta)}{p(\mathbf{X})}$  is intractable. We define the conditional probability of a parameter  $\Theta_k$  as follows

$$p(\Theta_k | \mathbf{X}_k, \Theta_{-k}) = \frac{p(\mathbf{X}_k | \Theta_k, \Theta_{-k}) p(\Theta_k)}{\int p(\mathbf{X}_k | \Theta_k, \Theta_{-k}) p(\Theta_k) d\Theta_k}$$

**NOTE** Here,  $\Theta_k$  is the set of all parameters excluding  $\Theta_k$  and  $\mathbf{X}_k$  is the data that is dependent on  $\Theta_k$

Suppose the conditional posteriors (CP) admit **local conjugacy** i.e. the above term is tractable for all parameters. Such models are called locally conjugate models.

#### 1.2 Bayesian Matrix Factorization

We assume the data to be modelled as a low rank matrix with some noise term. We need to factorize this  $N \times M$  matrix ( $\mathbf{R}$ ) into two matrices (interpreted as users and items)  $\mathbf{U}$  and  $\mathbf{V}$  of sizes  $N \times K$  and  $K \times M$  respectively such that  $K \ll N, M$  and  $\mathbf{R} \approx \mathbf{U} \times \mathbf{V}$

We assume the error term, and hence the likelihood to be gaussian

$$\begin{aligned} \mathbf{R} &= \mathbf{UV} + \epsilon \\ r_{ij} &= \mathbf{u}_i^T \mathbf{v}_j + \epsilon_{ij} \\ p(r_{ij}) &= \mathcal{N}(r_{ij} | \mathbf{u}_i^T \mathbf{v}_j, \beta^{-1}) < ++ > \end{aligned}$$

We assume Gaussian priors on the user and item latent features

$$\begin{aligned} p(\mathbf{u}_i) &= \mathcal{N}(\mathbf{u}_i | \mathbf{0}, \lambda_u^{-1} \mathbf{I}_K) \\ p(\mathbf{v}_j) &= \mathcal{N}(\mathbf{v}_j | \mathbf{0}, \lambda_v^{-1} \mathbf{I}_K) \end{aligned}$$

The BMF model with Gaussian likelihood and Gaussian priors has local conjugacy. We can write the posteriors for the latent variables as follows

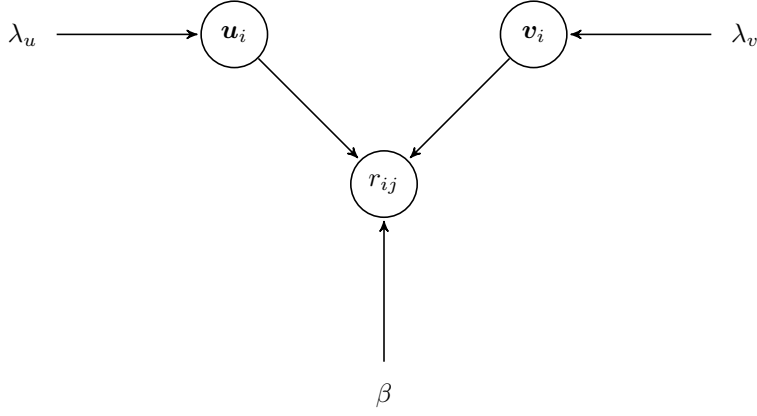


Figure 1: Matrix Factorization as Latent Factor Model

$$p(\mathbf{u}_i | \mathbf{R}, \mathbf{V}, \mathbf{U}_{-i}) = \prod_{j:r_{ij} \neq 0} p(r_{ij} | \mathbf{u}_i, \mathbf{v}_j) p(\mathbf{u}_i)$$

$$p(\mathbf{u}_j | \mathbf{R}, \mathbf{V}, \mathbf{U}_{-j}) = \prod_{i:r_{ij} \neq 0} p(r_{ij} | \mathbf{u}_i, \mathbf{v}_j) p(\mathbf{v}_j)$$

This is very similar to bayesian linear regression and we can find compute the posterior which is of the form

For user latent variables

$$p(\mathbf{u}_i | \mathbf{R}, \mathbf{V}) = \mathcal{N}(\mathbf{u}_i | \boldsymbol{\mu}_{\mathbf{u}_i}, \boldsymbol{\Sigma}_{\mathbf{u}_i})$$

$$\boldsymbol{\Sigma}_{\mathbf{u}_i} = \left( \lambda_u \mathbf{I}_K + \beta \sum_{j:r_{ij} \neq 0} \mathbf{v}_j \mathbf{v}_j^T \right)^{-1}$$

$$\boldsymbol{\mu}_{\mathbf{u}_i} = \boldsymbol{\Sigma}_{\mathbf{u}_i} \left( \beta \sum_{j:r_{ij} \neq 0} r_{ij} \mathbf{v}_j \right)$$

For item latent variables

$$p(\mathbf{v}_j | \mathbf{R}, \mathbf{U}) = \mathcal{N}(\mathbf{v}_j | \boldsymbol{\mu}_{\mathbf{v}_j}, \boldsymbol{\Sigma}_{\mathbf{v}_j})$$

$$\boldsymbol{\Sigma}_{\mathbf{v}_j} = \left( \lambda_v \mathbf{I}_K + \beta \sum_{i:r_{ij} \neq 0} \mathbf{u}_i \mathbf{u}_i^T \right)^{-1}$$

$$\boldsymbol{\mu}_{\mathbf{v}_j} = \boldsymbol{\Sigma}_{\mathbf{v}_j} \left( \beta \sum_{i:r_{ij} \neq 0} r_{ij} \mathbf{u}_i \right)$$

This idea is similar in spirit to alternating optimization methods (such as EM).

## 2 Sampling from Distributions

We can approximate a distribution using a set of randomly drawn samples from it. However, it is not possible to directly sample *difficult* distributions. Hence, we need sampling techniques that allow us to sample from

such distributions *e.g.* we sometimes might need to approximately infer / sample intractable posteriors. These samples can be used to approximate anything that depends on these distributions *e.g.* expectations or posteriors.

## 2.1 Empirical Distribution

We sample using  $L$  points  $\{\mathbf{z}^{(l)}\}_{l=1}^L$  and the empirical distribution is defined as

$$p_L(L) = \sum_{l=1}^L w_l \delta_{\mathbf{z}^{(l)}}(A)$$

where  $\delta$  is the Dirac Delta function

$$\delta_{\mathbf{z}}(A) = \begin{cases} 0 & \text{if } \mathbf{z} \in A \\ 1 & \text{if } \mathbf{z} \notin A \end{cases}$$

$w_l$  is the weight of a point. This distribution can be viewed as a histogram and the weight as the height of the histogram bar. This method can be used to sample from both simple as well as difficult distributions.

## 2.2 Transformation Methods

Transformation methods are used to sample (mostly) generic or standard distributions, using simple distributions which can be easily sampled (such as uniform distribution). We essentially transform

## 2.3 Inverse CDF Method

This method uses the change of variable rule to transform a simple distribution to more complex distributions. Suppose we want to sample the random variable  $z$  using a simpler distribution  $x$ , then we can write

$$p(z) = p(x) \left| \frac{\delta(x)}{\delta(z)} \right|$$

if  $x$  is a uniform distribution

$$x = \int_{-\infty}^{\hat{z}} p(z) dz = F(\hat{z})$$

Thus, we can first draw a random sample  $x$  from  $Unif(0, 1)$  and transform into  $z$  using  $\hat{z} = h^{-1}(u)$

## 2.4 Box-Muller Method

This is a transformation technique to generate samples from two-dimensional Gaussian Distribution. The transformation is from the distribution  $Unif^2(0, 1)$  to the desired one.

Algorithm 1: Box-Muller Transformation

1. Assume two samples  $u, v$  drawn from  $U \sim Unif(0, 1)$  and  $V \sim Unif(0, 1)$
2. Transform these random variables into  $R, \Theta$  such that  $R = \sqrt{-2 \log U}$  and  $\Theta = 2\pi V$
3. Since  $\Theta$  is a linear transformation, we know  $\Theta \sim Unif(0, 2\pi)$
4. For  $R$

$$\begin{aligned} p(R \leq r) &= p\left(\sqrt{-2 \log U} \leq r\right) \\ &= 1 - p\left(U < e^{-\frac{r^2}{2}}\right) \\ &= 1 - e^{-\frac{r^2}{2}} \end{aligned}$$

$$\implies f_R(r) = r e^{-\frac{r^2}{2}}$$

5. Since  $U$  and  $V$  are independent,  $R$  and  $\Theta$  will also be independent. Hence  $f_{R, \Theta}(r, \theta) = \frac{1}{2\pi} r e^{-\frac{r^2}{2}}$
6. We further transform these into two random variables  $Z_1$  and  $Z_2$

$$Z_1 = R \cos 2\pi\Theta \quad \text{and} \quad Z_2 = R \sin 2\pi\Theta$$

7. The join distribution of  $Z_1, Z_2$  is given by

$$\begin{aligned} f_{Z_1, Z_2}(z_1, z_2) \frac{\delta(z_1, z_2)}{\delta(r, \theta)} &= f_{R, \Theta}(r, \theta) \\ &= \left[ \frac{1}{\sqrt{2\pi}} e^{-\frac{z_1^2}{2}} \right] \left[ \frac{1}{\sqrt{2\pi}} e^{-\frac{z_2^2}{2}} \right] \\ &= \mathcal{N}(z_1, z_2 \mid \mathbf{0}, \mathbf{I}_2) \end{aligned}$$

It is also possible to transform Normal distribution to any other gaussian distribution. Suppose a random variable  $X \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_k)$ . Transform this into  $Z \sim \boldsymbol{\mu}, \boldsymbol{\Sigma}$  as follows

$$Z = \boldsymbol{\mu} + LX$$

where  $L$  is the Cholesky decomposition of  $\boldsymbol{\Sigma}$

**NOTE** *Cholesky decomposition of will always exist as it is a positive definite matrix*

## 2.5 Rejection Sampling

Suppose we have a random variable with an intractable normalization factor, we can still sample it using rejection sampling. Consider a distribution  $p(z)$  such that

$$p(z) = \frac{\tilde{p}(z)}{Z_p}$$

where  $Z_p$  is not computable.

In order to sample, we need a *proposal distribution*  $q(z)$  such that

$$Mq(z) > \tilde{p}(z) \quad \forall z \quad \text{where } M \text{ is a constant}$$

The steps to generate sample are as follows

### Algorithm 2: Rejection Sampling

1. Sample a random variable  $z_*$  from  $q(z)$
2. Sample a uniform random variable  $u \sim \text{Unif}(0, Mq(z_*))$
3. If  $u \leq \tilde{p}(z_*)$ , then *accept* else *reject*

The proof of the algorithm can be seen as follows

$$\begin{aligned} p(\text{accept} \mid z) &= \frac{\tilde{p}(z)}{Mq(z)} \\ p(z, \text{accept}) &= q(z) p(\text{accept} \mid z) = \frac{\tilde{p}(z)}{M} \\ p(\text{accept}) &= \int \frac{\tilde{p}(z)}{M} dz = \frac{Z_p}{M} \\ p(z \mid \text{accept}) &= \frac{\tilde{p}(z)}{Z_p} = p(z) \end{aligned}$$

<+++>

<+++>

<+++>