ALGORITHM FOR PMF BACKGROUND MODEL WITH WEIGHTS

ZIHAO WANG

1. Model.

$$X_{ij} = \sum_k Z_{ijk} \tag{1.1}$$

$$Z_{ijk} \sim \text{Pois}(w_k l_{i0} f_{j0} l_{ik} f_{jk}) \tag{1.2}$$

$$l_{ik} \sim g_{L,k}(.), f_{jk} \sim g_{F,k}(.) \tag{1.3}$$

We assume that X is sparse: it has m nonzero elements and $m << np$.

2. Notation.

For $v \sim q(.)$, I use

$$\bar{v} := E_q[v] \tag{2.1}$$

$$\hat{v} := exp(E_q[log(v)]) \tag{2.2}$$

And in the algorithm, q is always the corresponding variational distribution for that variable.

3. Algorithm.

3.1. ELBO.

$$\text{ELBO} = C_1 - C_2 + C3 \tag{3.1}$$

$$C_1 = E[log\ p(Z|L, F, L_0, F_0)] \tag{3.2}$$

$$= \sum_{ijk} (-w_k l_{i0} f_{j0} \bar{l}_{ik} \bar{f}_{jk} + \bar{Z}_{ijk} log(w_k l_{i0} f_{j0} \hat{l}_{ik} \hat{f}_{jk})) - \sum_{ijk} E(log(Z_{ijk}!)) \tag{3.3}$$

$$C_2 = E[log\ q_Z(Z)] \tag{3.4}$$

$$= \sum_{ijk} (\bar{Z}_{ijk} log(\zeta_{ijk})) + \sum_{ij} log(X_{ij}!) - \sum_{ijk} E(log(Z_{ijk}!)) \tag{3.5}$$

$$C_3 = E(log \frac{g_L(L)}{q_L(L)} + log \frac{g_F(F)}{q_F(F)}) \tag{3.6}$$

The last part of C_1, C_2 gets cancelled out.

For convenience , I introduce the following variable:

$$B_{ijk} := w_k \hat{l}_{ik} \hat{f}_{jk} \tag{3.7}$$

$$B_{ij} := \sum_k B_{ijk} \tag{3.8}$$

$$\Lambda_{ijk} := w_k \bar{l}_{ik} \bar{f}_{jk} \tag{3.9}$$

$$\Lambda_{ij} := \sum_k \Lambda_{ijk} \tag{3.10}$$

Remark. *In actual implementation: B_{ij} is computed and stored only for those where $X_{ij} \neq 0$; Λ_{ij} is not computed nor stored.*

3.2. Update formula. In iteration t

1. update $\bar{Z}_{ijk}$

   Using Lagrange Multipler on parts relevant to ζ, it's easy to see

$$\zeta_{ijk} \leftarrow \frac{B_{ijk}}{B_{ij}} \tag{3.11}$$

$$\bar{Z}_{ijk} = X_{ij}\zeta_{ijk} \tag{3.12}$$

   Remark. *We only need to compute ζ_{ijk} where $X_{ij} \neq 0$, and we don't need to store this 3 dimensional array. The memory and runtime are only $O(m)$, instead of $O(npK)$.*

2. update q_L, g_L, q_F, g_F

   By observing $C_1 + C_3$, we have

$$(q_{L,k}, g_{L,k}, \text{kl}_{L,k}) \leftarrow \text{EBPM}(y_i = \bar{Z}_{i.k}, s_i = (\sum_j f_{j0}\bar{f}_{jk})w_k l_{i0}) \tag{3.13}$$

$$(q_{F,k}, g_{F,k}, \text{kl}_{F,k}) \leftarrow \text{EBPM}(y_j = \bar{Z}_{.jk}, s_j = (\sum_i l_{i0}\bar{l}_{ik})w_k f_{j0}) \tag{3.14}$$

   Remark. *The computation of KL-divergence will be explained later.*

3. update B_{ij}

$$B_{ij} = B_{ij} - B_{ijk}^{(t-1)} + B_{ijk}^{(t)} \tag{3.15}$$

4. update l_{i0}, f_{j0}

   Take derivatives in C_1 and use the fact that $\sum_k \bar{Z}_{ijk} = X_{ij}$, we have

$$l_{i0} \leftarrow \frac{X_{i.}}{\sum_j f_{j0}\Lambda_{ij}} = \frac{X_{i.}}{\sum_k w_k \bar{l}_{ik}(\sum_j f_{j0}\bar{f}_{jk})} \tag{3.16}$$

$$f_{j0} \leftarrow \frac{X_{.j}}{\sum_i l_{i0}\Lambda_{ij}} = \frac{X_{j.}}{\sum_k w_k \bar{f}_{jk}(\sum_i l_{i0}\bar{l}_{ik})} \tag{3.17}$$

   Remark. *The computation is $O((n+p)K)$*

5. update w_k

   It is easy to see

$$w_k \leftarrow \frac{\sum_{ij} \bar{Z}_{ijk}}{\sum_{ij} l_{i0} f_{j0} \bar{l}_{ik} \bar{f}_{jk}} \tag{3.18}$$

3.3. ELBO computation after update. When we plug in the updates into the ELBO, we have

(3.19)

$$\text{ELBO} = \sum_{ij}(-l_{i0}f_{j0}\Lambda_{ij} + X_{ij}log(l_{i0}f_{j0}B_{ij}) - log(X_{ij}!)) + E(log\frac{g_L(L)}{q_L(L)} + log\frac{g_F(F)}{q_F(F)})$$

(3.20)

$$= \sum_{ij}(-l_{i0}f_{j0}\Lambda_{ij} + X_{ij}log(l_{i0}f_{j0}B_{ij}) - log(X_{ij}!)) - \sum_k \text{KL}(q_L(l_{Ik})|g_L(l_{Ik})) - \sum_k \text{KL}(q_F(f_{Jk})|g_F(f_{Jk}))$$

Remark. *Again, we don't need to compute Λ in actual computation. The KL-divergence will be explained in the next section.*

Algorithm 3.1 EBPMF-WBG

Input: $X_{IJ}, l_{I0}, f_{J0}, q_L, g_L, q_F, g_F, B_{IJ}$

for $t = 1, ..., T$ **do**

  KL $\leftarrow 0$

  for $k = 1, ..., K$ **do**

    $B_{ijk} \leftarrow w_k \hat{l}_{ik} \hat{f}_{jk}$

    $\bar{Z}_{ijk} \leftarrow X_{ij} \frac{B_{ijk}}{B_{ij}}$

    $(q_{L,k}, g_{L,k}, \text{kl}_{L,k}) \leftarrow \text{EBPM}(y_i = \bar{Z}_{i.k}, s_i = (\sum_j f_{j0} \bar{f}_{jk}) w_k l_{i0})$

    $(q_{F,k}, g_{F,k}, \text{kl}_{F,k}) \leftarrow \text{EBPM}(y_j = \bar{Z}_{.jk}, s_i = (\sum_i l_{i0} \bar{l}_{ik}) w_k f_{j0})$

    $w_k \leftarrow \frac{\sum_{ij} X_{ij}(B_{ijk}/B_{ij})}{(\sum_i l_{i0} \bar{l}_{ik})(\sum_j f_{j0} \bar{f}_{jk})}$

    $B_{ij} \leftarrow B_{ij} - B_{ijk} + w_k \hat{l}_{ik} \hat{f}_{jk}$

    KL $\leftarrow$ KL $+ \text{kl}_{F,k} + \text{kl}_{L,k}$

  end

  $l_{i0} \leftarrow \frac{X_{i.}}{\sum_k w_k \bar{l}_{ik} (\sum_j f_{j0} \bar{f}_{jk})}$

  $f_{j0} \leftarrow \frac{X_{j.}}{\sum_k w_k \bar{f}_{jk} (\sum_i l_{i0} \bar{l}_{ik})}$

  ELBO = compute-elbo(KL, X, B, Λ)

end

Output: $X_{IJ}, l_{I0}, f_{J0}, q_L, g_L, q_F, g_F$

3.4. Algorithm.

4. Compute KL divergence from EBPM. Consider the EBPM problem:

$$y_i \sim Pois(s_i \lambda_i) \tag{4.1}$$

$$\lambda_i \sim g(.) \tag{4.2}$$

Our procedure optimizes the marginal log-likelihood $\sum_i log\ p(y_i)$ for g^*, then compute the posterior $p(\lambda_i | y_i, g^*)$.

The objective can be re-writen this way:

$$\sum_i log\ p(y_i) = \sum_i log\ \frac{p(y_i|\lambda_i) g(\lambda_i)}{p(\lambda_i|y_i)} \tag{4.3}$$

$$= \sum_i E_{q(\lambda_i)}[log\ \frac{p(y_i|\lambda_i) g(\lambda_i)}{p(\lambda_i|y_i)}] \tag{4.4}$$

$$= \sum_i E_{q(\lambda_i)}[log\ p(y_i|\lambda_i)] - E_{q(\lambda_i)}[\frac{p(\lambda_i|y_i)}{g(\lambda_i)}] \tag{4.5}$$

Use $q(\lambda_i) := p(\lambda_i|y_i)$, we have

$$-\text{KL}(p(\boldsymbol{\lambda}|\boldsymbol{y})|g(\boldsymbol{\lambda})) = log\ p(\boldsymbol{y}) - \sum_i E_{p(\lambda_i|y_i)}[log\ p(y_i|\lambda_i)] \tag{4.6}$$

First term on RHS is given by EBPM; the second term can be easily computed.

5. Numerical Trick. Directly computing B_{ij} has the issue of overflows and underflows.

Let $a_{ij} := \max_k log(B_{ijk})$, and compute the following:

$$b_{ijk} := log(B_{ijk}) - a_{ij} \tag{5.1}$$

$$b_{ij} := log \sum_k \exp(b_{ijk}) \tag{5.2}$$

We can recover the needed quanities with b:

$$\frac{B_{ijk}}{B_{ij}} = \exp(b_{ijk} - b_{ij}) \tag{5.3}$$

$$log(B_{ij}) = b_{ij} + a_{ij} \tag{5.4}$$

The advantage of using b instead of B is that computing $exp(b_{ijk})$ is more numerically stable than computing B_{ijk}.