

ALGORITHM FOR POISSON MATRIX FACTORIZATION

ZIHAO WANG

1. Model.

$$(1.1) \quad X_{ij} = \sum_k Z_{ijk}$$

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

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

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

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

$$(2.1) \quad \bar{v} := E_q[v]$$

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

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

3. Algorithm.

3.1. ELBO.

$$(3.1) \quad \text{ELBO} = C_1 - C_2 + C_3$$

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

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

$$(3.4) \quad C_2 = E[\log q_Z(Z)]$$

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

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

The last part of C_1, C_2 gets cancelled out.

For convenience, I introduce the following variable:

$$(3.7) \quad B_{ijk} := \hat{l}_{ik} \hat{f}_{jk}$$

$$(3.8) \quad B_{ij} := \sum_k B_{ijk}$$

$$(3.9) \quad \Lambda_{ijk} := \bar{l}_{ik} \bar{f}_{jk}$$

$$(3.10) \quad \Lambda_{ij} := \sum_k \Lambda_{ijk}$$

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 Multiplier on C_2 , it's easy to see

$$(3.11) \quad \zeta_{ijk} \leftarrow \frac{B_{ijk}}{B_{ij}}$$

$$(3.12) \quad \bar{Z}_{ijk} = X_{ij} \zeta_{ijk}$$

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

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

$$(3.14) \quad (q_{F,k}, g_{F,k}, \text{kl}_{F,k}) \leftarrow \text{EBPM}(y_j = \bar{Z}_{j,k}, s_i = (\sum_i l_{i0} \bar{l}_{ik}) f_{j0})$$

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

3. update B_{ij}

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

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

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

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

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

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

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

$$(3.18)$$

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

$$(3.19)$$

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

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-BG

Input: $X_{IJ}, l_{I0}, f_{J0}, q_L, g_L, q_F, g_F, B_{IJ}$ **for** $t = 1, \dots, T$ **do** $KL \leftarrow 0$ **for** $k = 1, \dots, K$ **do** $B_{ijk} \leftarrow \hat{l}_{ik} \hat{f}_{jk}$ $\bar{Z}_{ijk} \leftarrow X_{ij} \frac{B_{ijk}}{B_{ij}}$ $(q_{L,k}, g_{L,k}, kl_{L,k}) \leftarrow \text{EBPM}(y_i = \bar{Z}_{i.k}, s_i = (\sum_j f_{j0} \bar{f}_{jk}) l_{i0})$ $(q_{F,k}, g_{F,k}, kl_{F,k}) \leftarrow \text{EBPM}(y_j = \bar{Z}_{.jk}, s_i = (\sum_i l_{i0} \bar{l}_{ik}) f_{j0})$ $B_{ij} \leftarrow B_{ij} - B_{ijk} + \hat{l}_{ik} \hat{f}_{jk}$ $KL \leftarrow KL + kl_{F,k} + kl_{L,k}$ **end** $l_{i0} \leftarrow \frac{X_{i.}}{\sum_k \bar{l}_{ik} (\sum_j f_{j0} \bar{f}_{jk})}$ $f_{j0} \leftarrow \frac{X_{.j}}{\sum_k \bar{f}_{jk} (\sum_i l_{i0} \bar{l}_{ik})}$ $\text{ELBO} = \text{compute-elbo}(KL, X, B, \Lambda)$ **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:

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

$$(4.2) \quad \lambda_i \sim g(\cdot)$$

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-written this way:

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

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

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

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

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

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