ANOTHER POISSON MEANS PROBLEM

ZIHAO WANG

1. Introduction. We want to fit this Poisson Matrix Factorization model:

$$\begin{aligned} X_{ij} &= \sum_k Z_{ijk} \\ Z_{ijk} &\sim Pois(l_{ik} f_{jk}) \\ f_{jk} &\sim g_{jk}(\mu_j, \phi_{jk}) \end{aligned}$$

Similar to what we have done before, the ELBO is

$$\begin{aligned} ELBO(q, g) &= E_q\{log\ p(Z|L, F) + +log\ \frac{g_F(F)}{q_F(F)} - log\ q_Z(Z)\} \\ &= \sum_{jk} E_{q_F}\{-l_{*k} + \langle Z_{*jk}\rangle_q log(f_{jk}) + log\frac{g_{jk}(f_{jk})}{q_{jk}(f_{jk})}\} - E_{q_Z}(log\ q_Z(Z)) \end{aligned}$$

Let's focus on f_{jk}. Let denote $s_k := l_{*k}$, $y_{jk} := \langle Z_{*jk}\rangle_q$, and replace f_{jk} with λ_{jk}, we can see $ELBO(q_F, g_F)$ is actually the ELBO of the following Poisson Means model

2. Poisson Means model.

$$\begin{aligned} y_{jk} &\sim Pois(s_k \lambda_{jk}) \\ \lambda_{jk} &\sim g_{jk}(\mu_j, \phi_{jk}) \end{aligned}$$

Suppose λ_{jk} is the scaled true expression for gene j of a cluster k. We assume it has mean μ_j (the background) and variance proportional to ϕ_{jk}. Ideally we want ϕ_{jk} to be sparse (I call it relative sparsity). Compare to previous "ebpm" model, we want few means to deviate from the background, instead of most of them being close to 0. (Think in terms of the EBPMF model, we want Φ to be sparse, instead of F).

We might approach this problem either through Empirical Bayes, or Hierarchical Bayes.

3. Gamma prior. It is natural to use this prior:

$$\begin{aligned} y_{jk} &\sim Pois(s_k \lambda_{jk}) \\ \lambda_{jk} &= \mu_j v_{jk} \\ v_{jk} &\sim Ga(1/\phi_{jk}, 1/\phi_{jk}) \end{aligned}$$

Note that I use the shape-rate parameterization for gamma: $E(v_{jk}) = 1$ and $Var(v_{jk}) = \phi_{jk}$.

It might be easier to write it this way

$$\begin{aligned} y_{jk} &\sim Pois(s_k \mu_j v_{jk}) \\ v_{jk} &\sim Ga(1/\phi_{jk}, 1/\phi_{jk}) \end{aligned}$$

Then it is easy to compute posterior:

$$v_{jk}|y_{jk} \sim Ga(1/\phi_{jk} + y_{jk}, 1/\phi_{jk} + s_k\mu_k)$$
$$\lambda_{jk} = \mu_j v_{jk}|y_{jk} \sim Ga(1/\phi_{jk} + y_{jk}, 1/\phi_{jk}\mu_j + s_k)$$

4. MLE, directly. It is a standard approach in EB to compute MLE w.r.t g. To compute the MLE (w.r.t μ_j, ϕ_{jk}), we integrate out v_{jk}:

$$y_{jk} \sim NB(1/\phi_{jk}, \frac{1}{1 + s_k\mu_j\phi_{jk}})$$

(as a sanity check $E(y_{jk}) = s_k\mu_j$). Then the log-likelihood is

$$\begin{aligned} l(y_{JK}|\mu_J, \phi_{JK}) = \sum_{j,k} \{ & log\Gamma(y_{jk} + 1/\phi_{jk}) - log\Gamma(1/\phi_{jk}) \\ & - (y_{jk} + 1/\phi_{jk})log(1 + s_k\mu_j\phi_{jk}) + y_{jk}(log(\mu_j) + log(\phi_{jk}))\} \end{aligned}$$

This seems a hard optimization problem. It is non-convex w.r.t ϕ_{jk}, and we can't get analytic form for $\mu_j^*(\phi_{jk})$ either, as

$$0 = \frac{\partial l}{\partial \mu_j} = \sum_k \frac{y_{jk} - s_k\mu_j}{\mu_j(1 + s_k\mu_j\phi_{jk})}$$

So we have to use coordinate descent, and in each iteration, we need to solve some non-convex problems.

5. MLE through EM. Use the following formulation (let $a_{jk} := 1/\phi_{jk}$)

$$y_{jk} \sim Pois(s_k\mu_j v_{jk})$$
$$v_{jk} \sim Ga(a_{jk}, a_{jk})$$

E-step is easy. Just compute the posterior

$$p(v|y, s, \mu, a) = Ga(a_{jk} + y_{jk}, a_{jk} + s_k\mu_j)$$

M-step: for one pair of (j, k), we can see μ and a are separated:

$$\begin{aligned} E[log\ p(y, v|a, \mu)] = & alog(a) - log\Gamma(a) + a(\langle log(v)\rangle - \langle v\rangle) \\ & - s\langle v\rangle \mu + ylog(\mu) + const \end{aligned}$$

Thus

$$\begin{aligned} J(\Theta, \Theta^{(\text{old})}) &= \sum_{jk} E[log\ p(y_{jk}, v_{jk}|a_{jk}, \mu_j)] \\ &= \sum_{jk} \{a_{jk}log(a_{jk}) - log\Gamma(a_{jk}) + a_{jk}(\langle log(v_{jk})\rangle - \langle v_{jk}\rangle)\} \\ &+ \sum_{jk} \{-s_k\langle v_{jk}\rangle \mu_j + y_{jk}log(\mu_j)\} \end{aligned}$$

Update of μ_j: (which makes sense intuitively...)

$$\mu_j \longleftarrow \frac{\sum_k y_{jk}}{\sum_k s_k \langle v_{jk} \rangle}$$

Update of a_{jk}: (though not closed form, it seems easy)

$$\begin{aligned} J(a_{jk}) &= a_{jk} log(a_{jk}) - log\Gamma(a_{jk}) + a_{jk}(\langle log(v_{jk}) \rangle - \langle v_{jk} \rangle) \\ J^{(1)}(a_{jk}) &= log(a_{jk}) - \psi^{(0)}(a_{jk}) + \langle log(v_{jk}) \rangle - \langle v_{jk} \rangle + 1 \\ J^{(2)}(a_{jk}) &= \frac{1}{a_{jk}} - \psi^{(1)}(a_{jk}) \end{aligned}$$

Note that the hessian is < 0 when a is not too big. When $a \longrightarrow \infty$, it seems to be approaching 0 from below. So it seems $J(a)$ is concave, so minimizing $-J(a)$ would be a convex problem.

5.1. Implementation. For the update of $a_{1:J,k}$ where $J = p$ can be huge, I formulate it as an optimization over a vector (the Hessian being a diagonal matrix).

I first tried to use the CVXR. But it doesn't recognize my objective function as convex (it seems to not support psigamma function in R).

So I just used nlm instead. It is faster when given gradient, and not Hessian.