Appendix to “Scalable Collaborative Bayesian Preference Learning”

Mohammad Emtiyaz Khan **Young Jun Ko** **Matthias Seeger**
École Polytechnique Fédérale de Lausanne, Switzerland

1 The standard EM is intractable

We show that the standard EM is intractable for the matrix factorization model. We make certain simplifications since our sole purpose is to demonstrate intractability. We assume a matrix factorization model without an offset, i.e. $\boldsymbol{z}_n = \boldsymbol{V}\boldsymbol{u}_n$. Similar to Takacs and Tikk [21], we assume Gaussian likelihoods on observations as shown below,

$$
\begin{aligned}
&\log p(\mathbb{O}_n|\boldsymbol{V},\boldsymbol{u}_n) && (1)\\
&= \sum_{ij\in\mathbb{O}_n} \left[-\tfrac{1}{2}\pi_{ijn}(\boldsymbol{v}_i^T\boldsymbol{u}_n - \boldsymbol{v}_j^T\boldsymbol{u}_n - y_{ijn})\right]^2 && (2)\\
&= -\tfrac{1}{2}(\tilde{\boldsymbol{z}}_n - \boldsymbol{y}_n)^T\boldsymbol{\Pi}_n(\tilde{\boldsymbol{z}}_n - \boldsymbol{y}_n) && (3)
\end{aligned}
$$

where $\tilde{\boldsymbol{z}}_n = \boldsymbol{B}_n\boldsymbol{z}_n$ as before, $\boldsymbol{\Pi}_n = \mathrm{diag}(\boldsymbol{\pi}_n)$, and $\boldsymbol{y}_n$ is a vector of y_{ijn}. We can fix y_{ijn} to 1 without loss of generality. Takacs and Tikk [21] make assumptions about π_{ijn} too, but we do not do that. In practice, an approximate inference method such as EP will be used to estimate π_{ijn} (and also y_{ijn}).

Consider an EM algorithm where $\boldsymbol{u}_n$ is integrated out in the E-step and $\boldsymbol{V}$ is estimated in the M-step. The M-Step criterion is shown below in terms of $\tilde{\boldsymbol{z}}_n$.

$$
\begin{aligned}
\phi(\boldsymbol{V}) &= \sum_{n=1}^{N} \mathbb{E}_{q(\boldsymbol{u}_n)}[-\log p(\mathbb{O}_n|\boldsymbol{V},\boldsymbol{u}_n)] && (4)\\
&= \sum_{n=1}^{N} \mathrm{tr}\left\{\boldsymbol{\Pi}_n\mathbb{E}_{q(\boldsymbol{u}_n)}\left[\tfrac{1}{2}(\tilde{\boldsymbol{z}}_n\tilde{\boldsymbol{z}}_n^T) - \boldsymbol{y}_n\tilde{\boldsymbol{z}}_n^T\right]\right\} && (5)
\end{aligned}
$$

We can express $\tilde{\boldsymbol{z}}_n$ in terms of $\boldsymbol{v} = \mathrm{vec}(\boldsymbol{V}^T)$,

$$
\tilde{\boldsymbol{z}}_n = \boldsymbol{B}_n\boldsymbol{V}\boldsymbol{u}_n = \boldsymbol{B}_n\left(\boldsymbol{I}_M \otimes \boldsymbol{u}_n^T\right)\boldsymbol{v}. \qquad (6)
$$

using which we see that $\phi(\boldsymbol{V})$ is quadratic in $\boldsymbol{v}$ and can be written as $\boldsymbol{v}^T\boldsymbol{A}\boldsymbol{v} + \boldsymbol{a}^T\boldsymbol{v}$. The complicated structure of matrix $\boldsymbol{A}$ can be seen in the following expression:

$$
\begin{aligned}
&\sum_{n=1}^{N} \mathrm{tr}\{\boldsymbol{\Pi}_n\mathbb{E}_{q(\boldsymbol{u}_n)}[\tfrac{1}{2}\tilde{\boldsymbol{z}}_n\tilde{\boldsymbol{z}}_n^T]\} = && (7)\\
&\tfrac{1}{2}\boldsymbol{v}^T\sum_{n=1}^{N}\mathbb{E}_{q(\boldsymbol{u}_n)}\left[(\boldsymbol{I}_M\otimes\boldsymbol{u}_n)\boldsymbol{B}_n^T\boldsymbol{\Pi}_n\boldsymbol{B}_n\left(\boldsymbol{I}_M\otimes\boldsymbol{u}_n^T\right)\right]\boldsymbol{v}
\end{aligned}
$$

In the case of explicit ratings modeled with isotropic Gaussian likelihoods, $\boldsymbol{B}_n^T\boldsymbol{\Pi}_n\boldsymbol{B}_n$ is diagonal and independent of n. Hence, the system will be block diagonal and can be solved independently for each row of $\boldsymbol{V}$. For pairwise or higher order interactions, $\boldsymbol{B}_n^T\boldsymbol{\Pi}_n\boldsymbol{B}_n$ will have off-diagonal entries, which, summed over all users will lead to a dense system matrix of size $MD \times MD$, far beyond reasonable memory capacities for large M and even just moderate D. Moreover, solving directly for $\boldsymbol{v}$ is out of the question. We can run a block coordinate descent algorithm, but if $\boldsymbol{A}$ is not stored, a new run over all users is required after each step.

2 Numerically safe computation

The following algorithm can be used for a numerically safe implementation of $\boldsymbol{e}_n, \boldsymbol{E}_n$.

Algorithm 1 Safe computation of $\boldsymbol{E}_n$ and $\boldsymbol{e}_n$

```
Require: $\boldsymbol{\Sigma}, \boldsymbol{\mu}, \boldsymbol{B}_n$
Ensure: $\boldsymbol{E}_n$ and $\boldsymbol{e}_n$
  $\boldsymbol{\Pi}_n \leftarrow \mathtt{diag}(\boldsymbol{\pi}_n)$
  $\boldsymbol{M}_n \leftarrow \boldsymbol{B}_n\boldsymbol{\Sigma}\boldsymbol{B}_n^T$
  $\tilde{\boldsymbol{L}}_n \leftarrow \mathtt{chol}(\boldsymbol{I} + \boldsymbol{\Pi}_n^{1/2}\boldsymbol{M}_n\boldsymbol{\Pi}_n^{1/2})$
  $\boldsymbol{E}_n \leftarrow \boldsymbol{\Pi}_n^{1/2}(\tilde{\boldsymbol{L}}_n\tilde{\boldsymbol{L}}_n{}^T)^{-1}\boldsymbol{\Pi}_n^{1/2}$
  $\boldsymbol{e}_n \leftarrow \boldsymbol{\Pi}_n^{1/2}(\tilde{\boldsymbol{L}}_n\tilde{\boldsymbol{L}}_n{}^T)^{-1}\left[\boldsymbol{\Pi}_n^{-1/2}\boldsymbol{\beta}_n - \boldsymbol{\Pi}_n^{1/2}\boldsymbol{B}_n\boldsymbol{\mu}\right]$
```

3 Predictive probabilities

Suppose that we want to predict the probability of unobserved pairs $i \succ j$ for a user n. If $\boldsymbol{b}_*^T$ denotes the corresponding coupling vector and $\tilde{z}_* = \boldsymbol{b}_*^T\boldsymbol{z}_n$, then we can compute mean and variance of $q(\tilde{z}_*)$ to get the predictive probability $q(i \succ j|\mathcal{D}) = \int \Phi(\tilde{z}_*)q(\boldsymbol{z}_n|\mathcal{D})d\boldsymbol{z}_n$.

$$
\begin{aligned}
\mathbb{E}(\tilde{z}_*^2) &= \boldsymbol{b}_*^T\boldsymbol{\Sigma}\boldsymbol{b}_* - \boldsymbol{b}_*^T\boldsymbol{\Sigma}\boldsymbol{B}_n\boldsymbol{E}_n\boldsymbol{B}_n^T\boldsymbol{\Sigma}\boldsymbol{b}_* && (8)\\
\mathbb{E}(\tilde{z}_*) &= \boldsymbol{b}_*^T\boldsymbol{\mu} + \boldsymbol{b}_*^T\boldsymbol{\Sigma}\boldsymbol{B}_n^T\boldsymbol{e}_n && (9)\\
q(i \succ j|\mathcal{D}) &= \Phi\left(\frac{\mathbb{E}(\tilde{z}_*)}{\sqrt{\mathbb{E}(\tilde{z}_*^2)+1}}\right). && (10)
\end{aligned}
$$

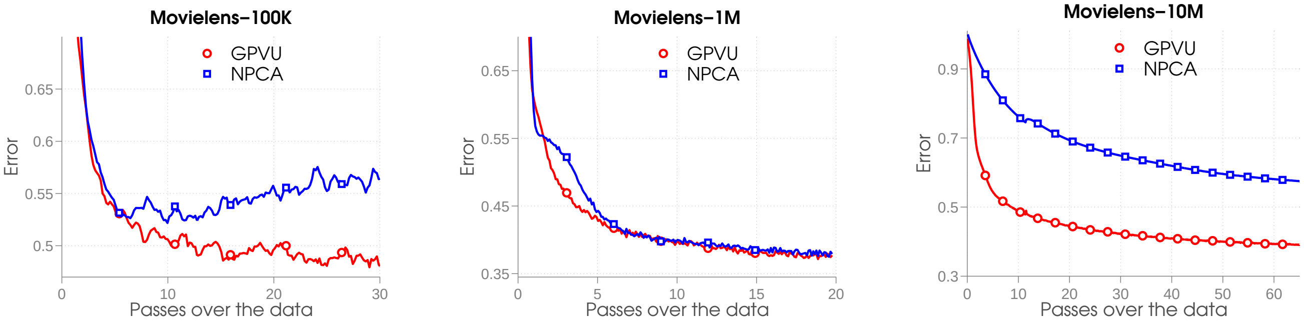

Figure 1: Comparison of NPCA-EP and GPVU-EP on MovieLens datasets (100k, 1M, 10M).

4 Expression for $\boldsymbol{S}$

We first plug-in the updated $\hat{\boldsymbol{\mu}}$, to get the following expression for $\boldsymbol{C}$:

$$\boldsymbol{C} = \frac{1}{N}\sum_{n=1}^{N} \mathrm{Cov}(\boldsymbol{z}_n) + [\hat{\boldsymbol{\mu}} - \mathbb{E}(\boldsymbol{z}_n)][\hat{\boldsymbol{\mu}} - \mathbb{E}(\boldsymbol{z}_n)]^T$$

We now simplify the second term. First, rewrite.

$$\hat{\boldsymbol{\mu}} - \mathbb{E}(\boldsymbol{z}_n) = \hat{\boldsymbol{\mu}} - \boldsymbol{\mu} + \boldsymbol{\mu} - \mathbb{E}(\boldsymbol{z}_n) \tag{11}$$

noting that $\boldsymbol{\mu}$ is the old value of the offset. The second term can be written as the sum of 4 terms.

$$(\hat{\boldsymbol{\mu}} - \boldsymbol{\mu})(\hat{\boldsymbol{\mu}} - \boldsymbol{\mu})^T + (\boldsymbol{\mu} - \mathbb{E}(\boldsymbol{z}_n))(\boldsymbol{\mu} - \mathbb{E}(\boldsymbol{z}_n))^T + \\ (\hat{\boldsymbol{\mu}} - \boldsymbol{\mu})(\boldsymbol{\mu} - \mathbb{E}(\boldsymbol{z}_n))^T + (\boldsymbol{\mu} - \mathbb{E}(\boldsymbol{z}_n))(\hat{\boldsymbol{\mu}} - \boldsymbol{\mu})^T \tag{12}$$

Using $\hat{\boldsymbol{\mu}} - \boldsymbol{\mu} = \boldsymbol{\Sigma s}$ and $\boldsymbol{\mu} - \mathbb{E}(\boldsymbol{z}_n) = -\boldsymbol{\Sigma B}_n^T \boldsymbol{e}_n$, we can simplify the above expression to the following.

$$\boldsymbol{\Sigma}(\boldsymbol{s}\boldsymbol{s}^T - \boldsymbol{s}\boldsymbol{e}_n^T\boldsymbol{B}_n - \boldsymbol{B}_n^T\boldsymbol{e}_n\boldsymbol{s}^T + \boldsymbol{B}_n^T\boldsymbol{e}_n\boldsymbol{e}_n^T\boldsymbol{B}_n)\boldsymbol{\Sigma} \tag{13}$$

Substituting this in the original expression for $\boldsymbol{C}$, taking the sum inside and using $\boldsymbol{s} = \sum_{n=1}^{N} \boldsymbol{B}_n\boldsymbol{e}_n/N$, we get the new expression for $\boldsymbol{C}$ and $\boldsymbol{S}$.

5 More results

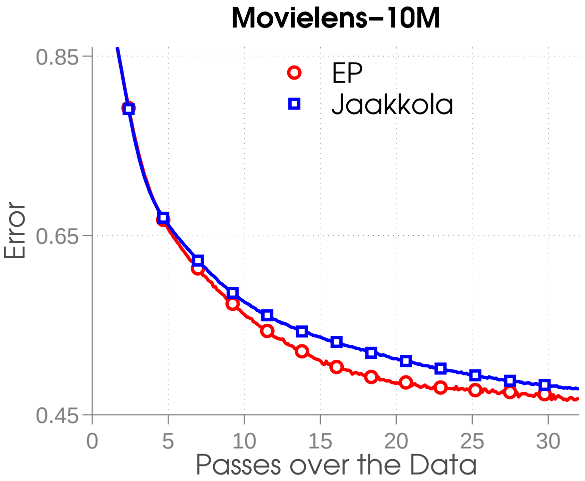

Figure 2: Comparison of GPVU-EP with GPVU-Jaakkola on MovieLens10M with 1M subsampled pairs. We show markers after every 20 iteration. EP performs better than Jaakkola.