Lab Report
Data & Knowledge - Factorization-Based Data Modeling
Practical Work 3

ZHANG Xin

January 2020

1 Develop a (non-negative) coupled factorization model for decomposing all these observed matrices/tensors simultaneously.

Suppose that we have $X_1 \in \mathbb{R}^{m\times n\times r}$, $X_2 \in \mathbb{R}^{m\times n}$, $X_3 \in \mathbb{R}^{n\times p}$, $X_4 \in \mathbb{R}^{m\times m}$, $X_5 \in \mathbb{R}^{r\times r}$.
So we adapt a PARAFAC-style approach, define a rank k, for the decomposition of X_1, we can have three low-dimensional representations:
User $U = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k] \in \mathbb{R}^{m\times k}$, Locations $L = [\mathbf{l}_1, \mathbf{l}_2, \dots, \mathbf{l}_k] \in \mathbb{R}^{n\times k}$ and Activities $A = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k] \in \mathbb{R}^{r\times k}$
And for X_3, in addition to L, we define the Feature $F = [\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_k] \in \mathbb{R}^{p\times k}$.
So we have:

$$\begin{aligned}
\hat{X}_1 &= [\![U, L, A]\!] = \Sigma_{i=1}^{k} \mathbf{u}_i \otimes \mathbf{l}_i \otimes \mathbf{a}_i \\
\hat{X}_2 &= UL^T \\
\hat{X}_3 &= LF^T \\
\hat{X}_4 &= UU^T \\
\hat{X}_5 &= AA^T
\end{aligned}$$

where $\otimes$ denotes the outer product.

2 Write down the cost function by using the β-divergence

The cost function

$$\begin{aligned}
\mathcal{L}(U, L, A, F) &= D_{\beta_1}(X_1 \| [\![U, L, A]\!]) + \lambda_2 D_{\beta_2}(X_2 \| UL^T) + \lambda_3 D_{\beta_3}(X_3 \| LF^T) + \lambda_4 D_{\beta_4}(X_4 \| UU^T) + \lambda_5 D_{\beta_5}(X_5 \| AA^T) \\
&=
\end{aligned}$$

3 Explain why the model makes sense.

The model makes sense because that when $X_{1:5} = \hat{X}_{1:5}$, the β-divergence $d_\beta(x\|\hat{x}) = \frac{x^\beta}{\beta(\beta-1)} - \frac{x\hat{x}^{\beta-1}}{\beta-1} + \frac{\hat{x}^\beta}{\beta} = 0$. So if we minimize the cost function defined in question 2 to 0, it will also minimize the difference between the model and the real dataset.

4 Develop the multiplicative update rules algorithm for the model

Here we try to apply gradient descend for the model:
We have:

$$\begin{aligned}
\nabla_U &= (\hat{X_1}^{\cdot(\beta_1-1)} - X_1 \cdot \hat{X_1}^{\cdot\beta_1-2})^{(1)}(A * L) + \lambda_2(\hat{X_2}^{\cdot(\beta_2-1)} - X_2 \cdot \hat{X_2}^{\cdot\beta_2-2})L + \lambda_4(\hat{X_4}^{\cdot(\beta_4-1)} - X_4 \cdot \hat{X_4}^{\cdot\beta_4-2})U \\
\nabla_L &= (\hat{X_1}^{\cdot(\beta_1-1)} - X_1 \cdot \hat{X_1}^{\cdot\beta_1-2})^{(2)}(A * U) + \lambda_2(\hat{X_2}^{\cdot(\beta_2-1)} - X_2 \cdot \hat{X_2}^{\cdot\beta_2-2})^T U + \lambda_3(\hat{X_3}^{\cdot(\beta_3-1)} - X_3 \cdot \hat{X_3}^{\cdot\beta_3-2})F \\
\nabla_A &= (\hat{X_1}^{\cdot(\beta_1-1)} - X_1 \cdot \hat{X_1}^{\cdot\beta_1-2})^{(3)}(L * U) + \lambda_5(\hat{X_5}^{\cdot(\beta_5-1)} - X_5 \cdot \hat{X_5}^{\cdot\beta_5-2})A
\end{aligned}$$

$$\nabla_F = \lambda_3(\hat{X_3}^{\cdot(\beta_3-1)} - X_3 \cdot \hat{X_3}^{\cdot\beta_3-2})^T L$$

So the update rule is:

$$U_{t+1} = U_t - \gamma\nabla_U$$
$$L_{t+1} = L_t - \gamma\nabla_L$$
$$A_{t+1} = A_t - \gamma\nabla_A$$
$$F_{t+1} = F_t - \gamma\nabla_F$$

5 Implement your algorithm in Octave. Monitor the overall cost function. What are the effect of choosing different β for each tensor? When the algorithm converges, check whether the individual model predictions $\hat{X}_{1:5}$ are close to the original tensors or not.