Semi-supervised Tensor Factorization

Jiaxin Hu

Email:jhu267@wisc.edu

Campus ID: 9080815732

1 Model

- Consider a binary tensor $Y \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ and a covaraite matrix $A \in \mathbb{R}^{d_1 \times k}$ on the first mode.

- First consider the logistic tensor regression model without the covariate:

$$
\begin{aligned}
&Assume\ Y_{ijk} \sim Ber(p_{ijk})\ independently;\\
&G = logit(\mathbb{E}Y) = logit(p_{ijk}) = log\frac{p_{ijk}}{1-p_{ijk}}\\
&G = U \times_1 M_1 \times_2 M_2 \times_3 M_3
\end{aligned}
$$

  where U is core tensor, M_1, M_2, M_3 are factor matrices. $U \in \mathbb{R}^{r_1 \times r_2 \times r_3}, M_1 \in \mathbb{R}^{d_1 \times r_1}, M_2 \in \mathbb{R}^{d_2 \times r_2}, M_3 \in \mathbb{R}^{d_3 \times r_3}$.

- There are three ways to add the covariate matrix into the model.

- First, consider the covariate matrix as a factor matrix replacing the M_1:

$$G = U \times_1 A \times_2 M_2 \times_3 M_3$$

  where U should change it dimension to $k \times r_2 \times r_3$.

  Strength: This method directly show the effect from the covariate matrix and use this method can reduce the computation of the model because we only need to upgrade U, M_2, M_3.

  Weakness: This method is in some way more "supervised" than other methods, which implies we may lose more information because we fix the first mode. And when A is a binary matrix or matrix with groups of high correlated columns, we may meet problems like over-parameter when estimate other factor matrices and core tensor.

- Second, add another term in the model:

$$G = U \times_1 M_1 \times_2 M_2 \times_3 M_3 + \tilde{A}\gamma$$

where $\gamma \in \mathbb{R}^{k\times 1}$ and $\tilde{A} \in \mathbb{R}^{d_1\times d_2\times d_3}$, $(\tilde{A}\gamma)[i,,] = A[i,]\gamma$. In a more precise way, the model can be:

$$G[i,j,k] = \sum_{i',j',k'} U[i',j',k']M_1[i,i']M_2[j,j']M_3[k,k'] + \gamma_1 A[i,1] + \cdots + \gamma_k A[i,k]$$

Strength: In this method, we can keep the term of M_1 which can help us to describe the information on mode 1 that can't be described by the covariate matrix.
Weakness: In this method, we add another k parameters, which makes the model or flexible and need more computation. And in this method, the effect of covariate can not be as strong as the first method.

- Third, we can implement a linear regression on the first mode and take the covariate matrix as a response matrix:

$$G = U \times_1 M_1 \times_2 M_2 \times_3 M_3; \quad A = WM_1 + \epsilon$$

In this method, the effect of covariate should be shown in the loss function like :

$$\min_{U,M_1,M_2,M_3,W} \langle Y,\ U \times_1 M_1 \times_2 M_2 \times_3 M_3\rangle - \sum log(1 + exp(U \times_1 M_1 \times_2 M_2 \times_3 M_3))$$
$$+\alpha \|A - WM_1\|_F^2$$

Strength: In this method, we still keep all the terms in the tensor factorization and separate the linear regression step. And if A is a binary matrix, we can also do the glm to connect A and M_1.
Weakness: The effect from the covariate matrix is even smaller to whole model, for it only works when calculating the loss function. And this model can't consider the covariate matrix and the tensor factorization process together. And the parameter in this model is much more.

2 Upgrade Algorithm

In the data "dnoation.MAT", the covariate matrix is a binary matrix. Therefore, if we use the first model, we will meet the problem of over-parameter, which means we can not upgrade the elements of core tensor and factor matrices using glm for it would produce NA. And in this data, A has 111 columns and I think we can consider each column refers to a property of the country rather than general label. So I prefer to use the second model.

Below are the algorithm steps:

- **Initialize:** Use tucker to get $U^{(0)}, M_1^{(0)}, M_2^{(0)}, M_3^{(0)}$and set $\gamma = \mathbf{1}_k$

- **Upgrade $U^{(0)} \to U^{(1)}$:**

$$G[i,j,k] = \sum_{i',j',k'} U[i',j',k']M_1[i,i']M_2[j,j']M_3[k,k'] + \gamma_1 A[i,1] + \cdots + \gamma_k A[i,k]$$

  regard $\sum_{i=1}^{k} \gamma_i A[i,1]$ as offset of glm, and obtain the estimate of $U[i',j',k']$ to get $U^{(1)}$

- **Upgrade $M_3^{(0)} \to M_3^{(1)}$:**

$$set\ T = U \times_1 M_1 \times M_2$$

$$G[i,j,k] = \sum_{l=1}^{r_3} T[i,j,l]M_3[k,l] + \gamma_1 A[i,1] + \cdots + \gamma_k A[i,k]$$

  Still regard $\sum_{i=1}^{k} \gamma_i A[i,1]$ as offset in the model and obtain the estimate of $M_3[k,l]$ to get $M_3^{(1)}$.

- The process of upgrading M_2, M_1 would be the same with M_3.

- **Orthogonalize:** After each upgrading of core tensor and factor matrix, we should normalize or orthogonalize the factor matrices through SVD or Tucker:

$$X = U \times_1 M_1 \times M_2 \times M_3 = U^* \times_1 M_1^* \times M_2^* \times M_3^*$$

  where U, M_1, M_2, M_3 are the terms we have in the present step, and do the tucker to X to get a normalized result.

- **Upgrade $\gamma^{(0)} \to \gamma^{(1)}$:**

$$G[i,j,k] = \sum_{i',j',k'} U[i',j',k']M_1[i,i']M_2[j,j']M_3[k,k'] + \gamma_1 A[i,1] + \cdots + \gamma_k A[i,k]$$

now regard the term $\sum_{i',j',k'} U[i',j',k']M_1[i,i']M_2[j,j']M_3[k,k']$ as offset of the model and obtain the estimate of γ to get the $\gamma^{(1)}$.

- **Log-Likelihood function:** To calculate the log-likelihood value:

$$\begin{aligned} l(U, M_1, M_2, M_3, \gamma) =& \langle Y,\ U \times_1 M_1 \times_2 M_2 \times_3 M_3 + A^*\gamma \rangle \\ & - \sum log(1 + exp(U \times_1 M_1 \times_2 M_2 \times_3 M_3 + \tilde{A}\gamma)) \end{aligned}$$

where $\tilde{A} \in \mathbb{R}^{d_1 \times d_2 \times d_3},\ (\tilde{A}\gamma)[i,,] = A[i,]\gamma$.