Non-Linear Cross-Domain Collaborative Filtering via Hyper-Structure Transfer: Supplementary Materials

Yan-Fu Liu YFLIU@NETDB.CS.NTHU.EDU.TW
National Tsing Hua University, ROC

Cheng-Yu Hsu CYHSU@NETDB.CS.NTHU.EDU.TW
National Tsing Hua University, ROC

Shan-Hung Wu SHWU@CS.NTHU.EDU.TW
National Tsing Hua University, ROC

1. Objective Solving

With the following Eqs.

$$\tilde{\mathcal{B}}^{(k)}_{:,:,h} = \boldsymbol{A}^{(k)}\boldsymbol{\Phi}^{(k)}_h\boldsymbol{E}^{(k)\top} \tag{1}$$

and

$$\mathrm{proj}^{(k)}(\boldsymbol{B}) \quad = \boldsymbol{A}^{(k)}\boldsymbol{\Psi}^{(k)}\boldsymbol{E}^{(k)\top} \tag{2}$$

explained in the main text, we can employ the multiplicative gradient descent method (Ding et al., 2006) to update the $\tilde{\boldsymbol{U}}^{(k)} = [\boldsymbol{U}^{(k)}, \boldsymbol{0}]$, $\tilde{\boldsymbol{V}}^{(k)} = [\boldsymbol{V}^{(k)}, \boldsymbol{0}]$, $\boldsymbol{A}^{(k)}$, and $\boldsymbol{E}^{(k)}$ in each iteration. The update rules for Eq. (5) of the main text are given below:

$$\tilde{\boldsymbol{U}}^{(k)} \leftarrow \tilde{\boldsymbol{U}}^{(k)} \circ \sqrt{\frac{\boldsymbol{X}^{(k)}\tilde{\boldsymbol{V}}^{(k)}\boldsymbol{F}^{(k)\top}}{\tilde{\boldsymbol{U}}^{(k)}\boldsymbol{F}^{(k)}\tilde{\boldsymbol{V}}^{(k)\top}\tilde{\boldsymbol{V}}^{(k)}\boldsymbol{F}^{(k)\top}}}, \tag{3}$$

$$\tilde{\boldsymbol{V}}^{(k)} \leftarrow \tilde{\boldsymbol{V}}^{(k)} \circ \sqrt{\frac{\boldsymbol{X}^{(k)}\tilde{\boldsymbol{U}}^{(k)}\boldsymbol{F}^{(k)}}{\tilde{\boldsymbol{V}}^{(k)}\boldsymbol{F}^{(k)\top}\tilde{\boldsymbol{U}}^{(k)\top}\tilde{\boldsymbol{U}}^{(k)}\boldsymbol{F}^{(k)}}}, \tag{4}$$

where $\boldsymbol{F}^{(k)} = \boldsymbol{A}^{(k)}\boldsymbol{\Psi}^{(k)}\boldsymbol{E}^{(k)\top} = \mathrm{proj}^{(k)}(\boldsymbol{B})$.

$$\boldsymbol{A}^{(k)} \leftarrow \boldsymbol{A}^{(k)} \circ \sqrt{\frac{[\tilde{\boldsymbol{U}}^{(k)\top}\boldsymbol{X}^{(k)}\tilde{\boldsymbol{V}}^{(k)}\boldsymbol{E}^{(k)}\boldsymbol{\Psi}^{(k)}]}{[\tilde{\boldsymbol{U}}^{(k)\top}\tilde{\boldsymbol{U}}^{(k)}\boldsymbol{A}^{(k)}\boldsymbol{\Psi}^{(k)}\boldsymbol{E}^{(k)\top}\tilde{\boldsymbol{V}}^{(k)\top}\tilde{\boldsymbol{V}}^{(k)}\boldsymbol{E}^{(k)}\boldsymbol{\Psi}^{(k)}]}}, \tag{5}$$

$$\boldsymbol{E}^{(k)} \leftarrow \boldsymbol{E}^{(k)} \circ \sqrt{\frac{[\tilde{\boldsymbol{V}}^{(k)\top}\boldsymbol{X}^{(k)}\tilde{\boldsymbol{U}}^{(k)}\boldsymbol{A}^{(k)}\boldsymbol{\Psi}^{(k)}]}{[\tilde{\boldsymbol{V}}^{(k)\top}\tilde{\boldsymbol{V}}^{(k)}\boldsymbol{E}^{(k)}\boldsymbol{\Psi}^{(k)}\boldsymbol{A}^{(k)\top}\tilde{\boldsymbol{U}}^{(k)\top}\tilde{\boldsymbol{U}}^{(k)}\boldsymbol{A}^{(k)}\boldsymbol{\Psi}^{(k)}]}}, \tag{6}$$

where $\circ$ denotes the elemental-wise product, $\frac{[\cdot]}{[\cdot]}$ denotes the elemental-wise division, and $\sqrt{\cdot}$ denotes elemental-wise square root. The detailed steps are given in Algorithm 1. Note that we initiate the last columns of $\tilde{\boldsymbol{U}}^{(k)}$ and $\tilde{\boldsymbol{V}}^{(k)}$ by zero vectors, and because they are updated by element-wise multiplications, the last columns of $\tilde{\boldsymbol{U}}^{(k)}$ and $\tilde{\boldsymbol{V}}^{(k)}$ will remain zeros during each iteration.

2. More on Experiments

In this section, we give more details about our settings and conduct more experiments to further study the performance of MOTAR. Table 1 shows some statistics of our real datasets.

Datasets	DBLP	MovieLens
#users	180,640	69,878
#items	141,507	10,677
#rating events	1,495,081	10,000,054
Avg. #ratings/user	8.277	143.107
Avg. #ratings/item	10.565	936.598

Table 1: Statistics of the real datasets.

We validate that minimizing the MOTAR objective score does improve performance. Figure 1 shows the typical correlation between the objective score and MAE of MOTAR over real datasets. This justifies the validity of our MOTAR objective.

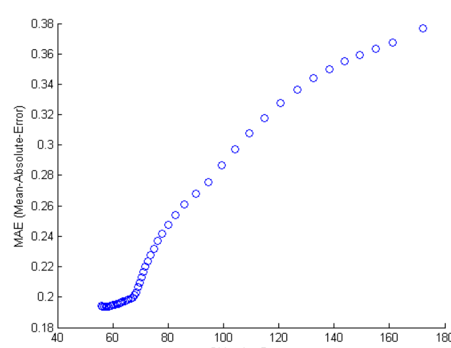

Figure 1: The correlation between the objective score and MAE of MOTAR.

Algorithm 1 The MOTAR training process.

Input: Dataset $\{\boldsymbol{X}^{(k)}\}_{1\leq k\leq d}$ and hyperparameters σ, β, $\{p^{(k)}, q^{(k)}\}_k$, and z
Output: $\{\boldsymbol{Y}^{(k)}\}_k$
Initialize $\{\tilde{\boldsymbol{U}}^{(k)}\}_k$ and $\{\tilde{\boldsymbol{V}}^{(k)}\}_k$ by random positives but set their last columns to $\mathbf{0}$
Initialize $\boldsymbol{\mathcal{B}}$ by random positives
repeat
  for $k \in \{1, \cdots, d\}$ **do**
    Obtain the cubicization $\tilde{\boldsymbol{\mathcal{B}}}^{(k)}$ from $\boldsymbol{\mathcal{B}}$, CP-decompose it by Eq. (1), and remember $\boldsymbol{\Phi}_h^{(k)}$'s
    Calculate $\mathrm{proj}^{(k)}(\boldsymbol{\mathcal{B}})$ by Eq. (2)
    Update $\tilde{\boldsymbol{U}}^{(k)}$, $\tilde{\boldsymbol{V}}^{(k)}$, $\boldsymbol{A}^{(k)}$, $\boldsymbol{E}^{(k)}$ by Eqs. (3)$\sim$(6)
    Normalize each row of $\tilde{\boldsymbol{U}}^{(k)}$, $\tilde{\boldsymbol{V}}^{(k)}$ by its l_1 norm
    Reconstruct $\boldsymbol{\mathcal{B}}$ by Eq.(1) using the remembered $\boldsymbol{\Phi}_h^{(k)}$'s
  end for
until convergence

References

Ding, Chris, Li, Tao, Peng, Wei, and Park, Haesun. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proc. of KDD*, 2006.