Frontiers of Information Technology & Electronic Engineering
www.zju.edu.cn/jzus; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Preference transfer model in collaborative filtering for implicit data*

Bin JU[1,2], Yun-tao QIAN‡[1], Min-chao YE[1]

([1]*Institute of Artificial Intelligence, College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China*)

([2]*Health Information Center of Zhejiang Province, Hangzhou 310006, China*)

E-mail: jubin_hz@163.com; ytqian@zju.edu.cn; yeminchao@126.com

Received Sept. 25, 2015; Revision accepted Feb. 17, 2016; Crosschecked May 14, 2016

Abstract: Generally, predicting whether an item will be liked or disliked by active users, and how much an item will be liked, is a main task of collaborative filtering systems or recommender systems. Recently, predicting most likely bought items for a target user, which is a subproblem of the rank problem of collaborative filtering, became an important task in collaborative filtering. Traditionally, the prediction uses the user item co-occurrence data based on users' buying behaviors. However, it is challenging to achieve good prediction performance using traditional methods based on single domain information due to the extreme sparsity of the buying matrix. In this paper, we propose a novel method called the preference transfer model for effective cross-domain collaborative filtering. Based on the preference transfer model, a common basis item-factor matrix and different user-factor matrices are factorized. Each user-factor matrix can be viewed as user preference in terms of browsing behavior or buying behavior. Then, two factor-user matrices can be used to construct a so-called 'preference dictionary' that can discover in advance the consistent preference of users, from their browsing behaviors to their buying behaviors. Experimental results demonstrate that the proposed preference transfer model outperforms the other methods on the Alibaba Tmall data set provided by the Alibaba Group.

Key words: Recommender systems, Collaborative filtering, Preference transfer model, Cross domain, Implicit data
http://dx.doi.org/10.1631/FITEE.1500313 **CLC number:** TP391

1 Introduction

Collaborative filtering (CF) is the mainstream technology for recommender systems. The main task in CF in general is to predict how much an item will be liked or disliked by active users. For instance, many previous CF studies have focused on the explicit types of data such as user-ratings data, e.g., a 1–5 score (Mnih and Salakhutdinov, 2007; Salakhutdinov and Mnih, 2008; Koren *et al.*, 2009; Koren, 2010). However, in real-world scenarios most feedback is not explicit but implicit. Implicit feedback is tracked automatically, such as by monitoring clicks, view times, purchases, and other user activities. Recently, predicting most likely bought items for a target user based on the implicit feedback data, which is a subproblem of the rank problem of CF, has become a hot research topic in CF (Rendle *et al.*, 2009; Shi *et al.*, 2012; Rendle and Freudenthaler, 2014).

Implicit feedback data is often modeled by the random triplet (U, I, F), where U is the user identifier, I is the item identifier, and F is the item browsing/buying frequency. Analogous to the construction of the user item co-occurrence matrix (the so-called 'rating matrix') in CF (Su and Khoshgoftaar,

‡ Corresponding author

* Project supported by the National Basic Research Program (973) of China (No. 2012CB316400) and the National Natural Science Foundation of China (No. 61571393)

ORCID: Bin JU, http://orcid.org/0000-0003-4709-4297

2009), we assemble the frequency of a user's browsing/buying preference of items into the matrix. Each element X_{ij} in the co-occurrence matrix indicates the number of times user j browses/buys item i. Given the user's past behavior of browsing/buying items, our task is finding which item will be most likely bought by the user in the next time period.

To tackle the prediction problem, the early memory-based methods of CF use the user's demographic information, such as gender, educational background, and hobby, to depict the user's profile of preference. The groups were made by identifying a set of similar users' preference profiles. Then, the item is recommended to the user who has never bought it, but which has been bought by another user in the same group. However, this type of profile is very difficult to obtain as it requires a large amount of extra information. Here, we are interested in building effective models by considering only the user item co-occurrence matrix.

From the user item co-occurrence data, we can cluster users into one or more latent user groups that reflect their buying preferences (Savia *et al.*, 2009). Knowing the user's buying preference in advance is a key point for the prediction. The accuracy of buying preference depends largely on the density of the given user buying co-occurrence matrix. However, in real-world recommender systems, users can buy a very limited number of items but browse a relatively large number of items. Thus, we ask: can we establish a bridge between the two domains and transfer useful preference patterns from the browsing matrix to the buying matrix?

One of the common shopping behaviors for people is the so-called 'look and then buy' approach. Generally, the more often a customer browses the items, the more likely the customer will buy the items. However, there is no linear relationship between the browsing frequency and the buying probability. By exploring the relationship between browsing frequency and buying probability using Alibaba's Tmall data, we observed that the possibility of the items being pursued declines after being browsed more than 50 times (Fig. 1). It indicates that if a customer likes to browse an item, it does not necessarily mean that the customer will buy it. In other words, the interest-driving factors behind the browsing of items and the buying of items for the customer might be different, and therefore we cannot use only the browsing frequency to predict the customer's buying behavior.

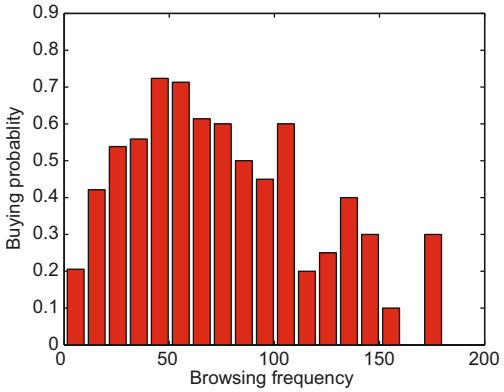

Fig. 1 Non-linear relationship between user's browsing and buying behaviors

In this study, we transfer user preference from one task to other related tasks to solve the prediction task. We name the prediction task the 'target task' and the related task the 'auxiliary task'. Suppose we have M items and N users. The target task is represented as a sparse $M \times N$ buying matrix $\boldsymbol{X}$, containing few observed buying data which would result in poor prediction results. Meanwhile, we obtain an auxiliary task from another domain, which is related to the target one and has a dense $M \times N$ browsing matrix $\boldsymbol{Y}$. Then, we adopt a variant of non-negative matrix factorization (NMF) to project user's item browsing or buying data from the original matrix into the preference space and to transfer preference from the browsing domain to the buying domain, which might enable better prediction results in the target task. We refer to this preference of patterns to be transferred as a preference transfer model (PTM), which is an $N \times K$ ($K \ll M, K \ll N$) user-factor matrix that factorizes from $\boldsymbol{X}$ and $\boldsymbol{Y}$. By capturing the consistency of preference patterns, we can reconstruct the target buying matrix and pick up the most likely recommended items to a certain user.

2 Related work

Recently, latent factor models have been successfully used in document modeling and data rating in CF (Hofmann and Puzicha, 1999; Si and Jin, 2003; Savia *et al.*, 2009). Most researchers have used latent factors to model the groups of similar users, which represent their preferences behind the users' behaviors. For instance, a probabilistic matrix

factorization (PMF) model was proposed where latent factors have been used to represent user's similar preference on rating items such as movies (Mnih and Salakhutdinov, 2007). Similar to grouping high-frequency words into different latent factors (topics), a probabilistic latent semantic analysis (PLSA) model was proposed to group similar users for predicting the unknown rating data (Hofmann, 2004). However, the static latent factors model cannot track the varying user's preference (Blei and Lafferty, 2006; Ju *et al.*, 2015). Furthermore, a joint PLSA probabilistic hypertext-induced topic selection (JPP) model was proposed to perform a simultaneous decomposition of the contingency tables associated with word occurrences and citations/links into a topic, which can bring forth cluster information, from relationships between documents to those inside documents (Cohn and Hofmann, 2000). However, one of the assumptions of the JPP—'the same latent factor in the cross domain'—is too hard. Because, in most cases, users' browsing and buying preferences are not necessarily required to be the same. Obviously, the assumptions need to be loosened. Unfortunately, a follow-up method for the JPP model does not appear in the literature for CF.

Another widely used approach to CF is NMF (Lee and Seung, 1999). It constitutes a rich body of algorithms that have found applications in a variety of machine learning fields, from document clustering to recommender systems (Zhang *et al.*, 2006; Chen *et al.*, 2009; Gu *et al.*, 2010). Historically, PLSA and NMF were developed independently, but researchers later proved that PLSA solves the problem of NMF with KL-divergence (Gaussier and Goutte, 2005; Ding *et al.*, 2006). Specifically, the non-negative constraint of NMF in the latent factors space can be used to model a user's preference. To tackle the one-side factor that JPP assumes, some researchers outlined a novel sentiment transfer mechanism based on constrained non-negative matrix trifactorizations of term document matrices in the source and target domains (Li *et al.*, 2010; Zhuang *et al.*, 2011; Xie *et al.*, 2012). Similarly, in CF, a little research has been conducted on rating data by transferring rating knowledge across multiple domains (Singh and Gordon, 2008; Li *et al.*, 2009a). Specifically, to pool together the rating data from multiple rating matrices in related domains for knowledge transfer and sharing, a rating-matrix generative model (RMGM) was proposed to learn latent user-cluster variables which can be viewed as a user's preference (Li *et al.*, 2009b).

We have not seen any method based on a dense auxiliary user item co-occurrence matrix in other domains to facilitate a sparse user item co-occurrence matrix in a target domain. Motivated by JPP's assumption of the same latent factor and RMGM's assumption of different latent factors, we hypothesize that projecting the user behavior into the latent factor space is more stable than in the original user item co-occurrence data space, and knowing the browsing preference in advance is more helpful for predicting buying preference for the next time period. Furthermore, inspired by Poisson matrix factorization (Ma *et al.*, 2011; Gopalan *et al.*, 2013), we propose a novel method of transferring user preference from the browsing matrix to the buying matrix by extending NMF.

3 Non-negative matrix factorization

In NMF, the goal is to find entrywise non-negative matrices $\boldsymbol{W}$ and $\boldsymbol{H}$ such that

$$(\boldsymbol{W},\boldsymbol{H})=\underset{\boldsymbol{W},\boldsymbol{H}}{\arg\min}\,\mathcal{L}\left(\boldsymbol{X}\|\boldsymbol{W}\boldsymbol{H}\right), \tag{1}$$

where $\boldsymbol{X}$ is the observation data matrix and $\mathcal{L}(\cdot)$ is a suitable loss function which can be the Euclidean distance or divergence (Lee and Seung, 1999; Devarajan *et al.*, 2015). Since $\boldsymbol{X}$ is fixed, we can use the divergence function

$$d(X_{ij},(\boldsymbol{W}\boldsymbol{H})_{ij})=-X_{ij}\log(\boldsymbol{W}\boldsymbol{H})_{ij}+(\boldsymbol{W}\boldsymbol{H})_{ij}, \tag{2}$$

and write the equivalent optimization problem with problem (1):

$$\underset{\boldsymbol{W},\boldsymbol{H}}{\arg\min}\sum_{i,j}d\left(X_{ij},(\boldsymbol{W}\boldsymbol{H})_{ij}\right). \tag{3}$$

In general, finding the global optimum cannot be guaranteed because this optimization problem is not convex with respect to both $\boldsymbol{W}$ and $\boldsymbol{H}$. However, the problem is convex with respect to $\boldsymbol{W}$ and $\boldsymbol{H}$ separately, which allows for the finding of locally optimal solutions.

Because of their computational effectiveness and simplicity, the multiplicative updates (MU) proposed in Lee and Seung (2000) are as follows:

$$W_{ik}\leftarrow W_{ik}\frac{\sum_j H_{kj}X_{ij}/(\boldsymbol{W}\boldsymbol{H})_{ij}}{\sum_j H_{kj}}, \tag{4}$$

$$H_{kj} \leftarrow H_{kj} \frac{\sum_i W_{ik} X_{ij} / (\boldsymbol{WH})_{ij}}{\sum_i W_{ik}}. \tag{5}$$

This alternative iterative algorithm had been proved to converge (Lee and Seung, 2000).

4 The proposed method

4.1 Preference transfer model

Here, we formulate the problem via NMF. Suppose $\boldsymbol{X} \in \mathbb{R}_+^{M \times N}$ is the buying matrix and $\boldsymbol{Y} \in \mathbb{R}_+^{M \times N}$ is the browsing matrix. An element $X_{ij} = x$ in the buying matrix represents user j buying x instances of item i. An element $Y_{ij} = y$ in the browsing matrix represents user i selecting y instances of item j. Because the data in the buying and browsing matrices comprises the occurrence of events in a period of time, $\boldsymbol{X}$ and $\boldsymbol{Y}$ are thought to conform to a Poisson distribution.

In our model, the buying matrix $\boldsymbol{X}$ is factorized into the item-factor matrix (or basis matrix) $\boldsymbol{W} \in \mathbb{R}_+^{M \times K}$ and the user-factor matrix (or coefficient matrix) $\boldsymbol{H} \in \mathbb{R}_+^{K \times N}$, where K represents the dimension of the latent factor space. The item-factor matrix $\boldsymbol{W}$ represents the projection from the item space to the latent factor space, whereas the user-factor matrix $\boldsymbol{H}$ represents the coefficients for a user's preferences for the corresponding factors (Lathia *et al.*, 2009). Furthermore, each column of $\boldsymbol{H}$, which is normalized, represents the distribution of user's preference in the latent factor space. The larger the element of a column of $\boldsymbol{H}$ is, the more probably the user's preference is dominated by the corresponding latent factor. Because $\boldsymbol{X}$ and $\boldsymbol{Y}$ are factorized simultaneously into the same factor space, the resulting user-factor matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ can be viewed as the user preference in terms of browsing behavior or buying behavior. Meanwhile, the property of items is relatively stationary, and $\boldsymbol{X}$ and $\boldsymbol{Y}$ are thought to share a common basis matrix $\boldsymbol{W}$. The probabilistic graph of PTM is shown in Fig. 2. Accordingly, we have the following:

$$X_{ij} \sim \mathrm{PO}\left(X_{ij}; (\boldsymbol{WU})_{ij}\right), \tag{6}$$

$$Y_{ij} \sim \mathrm{PO}\left(Y_{ij}; (\boldsymbol{WV})_{ij}\right), \tag{7}$$

where PO is the Poisson distribution, defined as follows:

$$\mathrm{PO}\left(x; \lambda\right) = \frac{\mathrm{e}^{-\lambda} \lambda^x}{\Gamma(x+1)}, \tag{8}$$

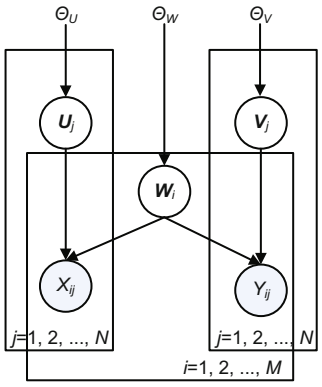

Fig. 2 Graphical model for preference transfer modeling

where $\Gamma(x)$ denotes the Gamma (generalized factorial) function which is defined as $\int_0^{+\infty} t^{x-1} \mathrm{e}^{-t} \mathrm{d}t$.

Assuming that the observed data is independent and identically distributed, we define the conditional distribution of the buying matrix as follows:

$$p\left(\boldsymbol{X} \mid \boldsymbol{W}, \boldsymbol{U}\right) = \prod_{i=1}^{M} \prod_{j=1}^{N} \mathrm{PO}\left(X_{ij}; (\boldsymbol{WU})_{ij}\right), \tag{9}$$

and the conditional distribution of the browsing matrix as follows:

$$p\left(\boldsymbol{Y} \mid \boldsymbol{W}, \boldsymbol{V}\right) = \prod_{i=1}^{M} \prod_{j=1}^{N} \mathrm{PO}\left(Y_{ij}; (\boldsymbol{WV})_{ij}\right). \tag{10}$$

We assume that the observations of $\boldsymbol{X}$ and $\boldsymbol{Y}$ are independent. Thus, we obtain the decomposition of the joint likelihood distribution:

$$p\left(\boldsymbol{X}, \boldsymbol{Y} \mid \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}\right) = p\left(\boldsymbol{X} \mid \boldsymbol{W}, \boldsymbol{U}\right) p\left(\boldsymbol{Y} \mid \boldsymbol{W}, \boldsymbol{V}\right). \tag{11}$$

We also assume that the prior factorizes as $p\left(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}\right) = p\left(\boldsymbol{U}\right) p\left(\boldsymbol{V}\right) p\left(\boldsymbol{W}\right)$. Because Gamma distribution is the conjugate prior to Poisson distribution, we further assume that the elements of $\boldsymbol{U}$, $\boldsymbol{V}$, and $\boldsymbol{W}$ are Gamma distributed with hyperparameters Θ_U, Θ_V, and Θ_W, respectively. Typically, we do not use many free hyperparameters and set them to be the same. Thus, we define the priori probability as follows:

$$\begin{aligned} &U_{ij} \sim G\left(U_{ij}; a_u, b_u\right), V_{ij} \sim G\left(V_{ij}; a_v, b_v\right), \\ &W_{ij} \sim G\left(W_{ij}; a_w, b_w\right), \end{aligned} \tag{12}$$

where $G(x; a, b)$ is the Gamma distribution for x with shape parameter a and scale parameter b. Based on the condition distribution and

the prior distribution, the joint posterior distribution is given by Bayes' rule $p(\boldsymbol{U},\boldsymbol{V},\boldsymbol{W}|\boldsymbol{X},\boldsymbol{Y}) \propto p(\boldsymbol{X},\boldsymbol{Y}|\boldsymbol{U},\boldsymbol{V},\boldsymbol{W})p(\boldsymbol{U},\boldsymbol{V},\boldsymbol{W})$, which factorizes to $p(\boldsymbol{X}|\boldsymbol{W},\boldsymbol{U})p(\boldsymbol{Y}|\boldsymbol{W},\boldsymbol{V})p(\boldsymbol{U})p(\boldsymbol{V})p(\boldsymbol{W})$.

The maximum a posteriori (MAP) state can be found as

$$\begin{aligned}\arg\max_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{W}} \big\{ &\log p(\boldsymbol{X}|\boldsymbol{W},\boldsymbol{U}) + \log p(\boldsymbol{Y}|\boldsymbol{W},\boldsymbol{V}) \\ &+ \log p(\boldsymbol{U}) + \log p(\boldsymbol{V}) + \log p(\boldsymbol{W}) \big\}.\end{aligned} \tag{13}$$

4.2 Algorithm

To solve Eq. (13), we substitute the terms in Eq. (13) with the expressions in Eqs. (2), (9), (10), and (12) and have the loss function as follows:

$$\begin{aligned}\mathcal{L} = &\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{W}} \Bigg[\sum_{i=1}^{M}\sum_{j=1}^{N} (-X_{ij}\log(\boldsymbol{W}\boldsymbol{U})_{ij} + (\boldsymbol{W}\boldsymbol{U})_{ij}) \\ &+ \sum_{i=1}^{M}\sum_{j=1}^{N} (-Y_{ij}\log(\boldsymbol{W}\boldsymbol{V})_{ij} + (\boldsymbol{W}\boldsymbol{V})_{ij}) \\ &+ \sum_{k=1}^{K}\sum_{j=1}^{N} \left(\frac{U_{kj}}{b_u} - (a_u-1)\log U_{kj} \right) \\ &+ \sum_{k=1}^{K}\sum_{j=1}^{N} \left(\frac{V_{kj}}{b_v} - (a_v-1)\log V_{kj} \right) \\ &+ \sum_{i=1}^{M}\sum_{k=1}^{K} \left(\frac{W_{ik}}{b_w} - (a_w-1)\log W_{ik} \right) \Bigg] \\ &\text{s.t.} \quad \sum_{k=1}^{K} U_{kj} = 1, \quad \sum_{k=1}^{K} V_{kj} = 1.\end{aligned} \tag{14}$$

Since the logarithm posterior probability is now written as a sum of the divergence functions, the MAP estimator can be derived directly by applying the MU (Lee and Seung, 2000) on the sum of the terms in Eqs. (15)–(17). To simplify the notation, we define $A_{ij} = X_{ij}/(\sum_{k=1}^{K} W_{ik}U_{kj})$ and $B_{ij} = Y_{ij}/(\sum_{k=1}^{K} W_{ik}V_{kj})$. The MU rule for each of the parameters is given as follows:

$$W_{ik} \leftarrow W_{ik} \frac{\sum_j U_{kj}A_{ij} + \sum_j V_{kj}B_{ij} + \dfrac{a_w-1}{W_{ik}}}{\sum_j U_{kj} + \sum_j V_{kj} + \dfrac{1}{b_w}}, \tag{15}$$

$$U_{kj} \leftarrow U_{kj} \frac{\sum_i W_{ik}A_{ij} + \dfrac{a_u-1}{U_{kj}}}{\sum_i W_{ik} + \dfrac{1}{b_u}}, \; U_{kj} \leftarrow \frac{U_{kj}}{\sum_k U_{kj}}, \tag{16}$$

$$V_{kj} \leftarrow V_{kj} \frac{\sum_i W_{ik}B_{ij} + \dfrac{a_v-1}{V_{kj}}}{\sum_i W_{ik} + \dfrac{1}{b_v}}, \; V_{kj} \leftarrow \frac{V_{kj}}{\sum_k V_{kj}}. \tag{17}$$

It can be seen that the update rules differ from the basic NMF updates only by the additive terms in the numerator and denominator, which are caused by the prior probabilities. According to the Appendix, the value of the loss function is guaranteed to be nondecreasing under each of the updates. To ensure the normalization of $\boldsymbol{U}$ and $\boldsymbol{V}$ without impact to nondecreasing under each of the updates, according to the method in Liu *et al.* (2013), we multiply $\sum_k U_{kj}$ and $\sum_k V_{kj}$ to each row of $\boldsymbol{W}$.

After $\boldsymbol{U}$ representing the user's buying preference and $\boldsymbol{V}$ representing the user's browsing preference were obtained, we can set the largest element of each column of $\boldsymbol{U}$ and $\boldsymbol{V}$ to be one and the others to be zero. Then the so-called 'preference dictionary' is constructed by performing $\boldsymbol{P} = \boldsymbol{U} \circledast \boldsymbol{V}$ ('$\circledast$' represents element-wise multiplication), which captures the consistency of users' preference from the buying matrix and the browsing matrix. Because recommending the most suitable item i to a certain user j is equivalent to choosing $\max_{i,j}(\sum_k W_{ik}P_{kj})$, we obtain the reconstructive matrix $\boldsymbol{R} = \boldsymbol{W}\boldsymbol{P}$. Moreover, it is greatly helpful to filter the 'noise' data in $\boldsymbol{R}$ with some rules. For example, we simply construct a rule function

$$F_{ij} = \delta(X_{ij}) = \begin{cases} 1, & \sum_j X_{ij} > \theta, \\ 0, & \text{otherwise}, \end{cases} \tag{18}$$

which indicates that item i ever bought more than θ times in the buying matrix $\boldsymbol{X}$ is regarded as a popular item. Subsequently, we can select the sorted elements in the filtering matrix $\boldsymbol{S} = \boldsymbol{R} \circledast \boldsymbol{F}$ with values greater than threshold τ to recommend. We summarize our method in Algorithm 1.

4.3 Complexity analysis

The computational cost of the algorithm is dominated by matrix factorization. In the PTM

model, Eq. (15) costs $O(2MNK)$ time, Eq. (16) costs $O(MNK)$ time, and Eq. (17) costs $O(MNK)$ time. In summary, the total time complexity of Algorithm 1 is $O(TMNK)$, where T is the number of iterations and is often set as a constant. Since our algorithm will converge after 5 to 10 iterations (see the details in experiments), this complexity analysis shows that our proposed approach is very efficient and can scale up with respect to very large datasets. The space cost of the algorithm is dominated by matrices $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{R}$, $\boldsymbol{F}$, $\boldsymbol{S}$, $\boldsymbol{U}$, $\boldsymbol{V}$, $\boldsymbol{W}$, and two temporary matrices. Therefore, the space complexity is $O(5MN + 6K(M + N))$.

Algorithm 1 Iterative algorithm for PTM

Input: Target matrix $\boldsymbol{X} \in \mathbb{R}_+^{M\times N}$, auxiliary matrix $\boldsymbol{Y} \in \mathbb{R}_+^{M\times N}$, dimension size K, hyperparameters $a_u, a_v, a_w, b_u, b_v, b_w$, popular θ, and threshold τ

// a_u, a_v, a_w should be set greater than 1 for $\boldsymbol{U}$, $\boldsymbol{V}$, and

// $\boldsymbol{W}$ to hold positive under each update

Output: Recommendation matrix $\boldsymbol{S} \in \mathbb{R}_+^{M\times N}$

1: Initialize $\boldsymbol{U}$, $\boldsymbol{V}$, and $\boldsymbol{W}$ with random numbers in $[0, 1]$

2: **repeat**

3: Fix $\boldsymbol{V}$ and $\boldsymbol{W}$, and update $\boldsymbol{U}$ with MU rule (16)

4: Fix $\boldsymbol{U}$ and $\boldsymbol{W}$, and update $\boldsymbol{V}$ with MU rule (17)

5: Fix $\boldsymbol{U}$ and $\boldsymbol{V}$, and update $\boldsymbol{W}$ with MU rule (15)

6: **until** the change of loss function (14) in this iteration is less than the threshold

7: The largest elements from each column of $\boldsymbol{U}$ and $\boldsymbol{V}$ are set to be 1 and others to be 0

8: $\boldsymbol{P} = \boldsymbol{U} \circledast \boldsymbol{V}$

9: $\boldsymbol{R} = \boldsymbol{W}\boldsymbol{P}$ // reconstructive matrix

10: $F_{ij} = \delta(X_{ij}) = \begin{cases} 1, & \sum\limits_j X_{ij} > \theta, \\ 0, & \text{otherwise.} \end{cases}$

// The popular items in $\boldsymbol{X}$ are picked out to mask the

// reconstructive matrix

11: $\boldsymbol{S} = \boldsymbol{R} \circledast \boldsymbol{F}$ // The sorted elements in the $\boldsymbol{S}$ whose values

// are greater than the threshold τ are recommended

5 Experiments

In this section, a simulation and a real experiment were conducted to demonstrate the effectiveness of the proposed PTM in predicting which items a user would buy in his/her next purchase based on his/her previous buying and browsing behaviors.

5.1 Simulation

Given the buying matrix of past time and the browsing matrix of the current time in a simulation, where the row represents the item and the column represents the user, we investigate the performance of our model. As shown in Fig. 3, after matrix factorization from step 1 (the dimension of the latent space is two), we obtain user preference matrices for both browsing and buying. After finding the element whose value is greater than 0.5 in both user-factor matrices, which means user preference is over 50% on some components of latent factors, we obtain the primary component matrix of user preference in the latent space. After performing the element-wise multiplication of the two matrices in step 3, which is based on the consistency of the main user preference in browsing behavior and buying behavior, the 'preference dictionary' is constructed. From the prediction matrix $\boldsymbol{P}$ reconstructed in step 4, we pick up the elements whose value is greater than 0.5, i.e., P_{51}, P_{13}, and P_{15}, which represent the items that have been visited and purchased frequently and may

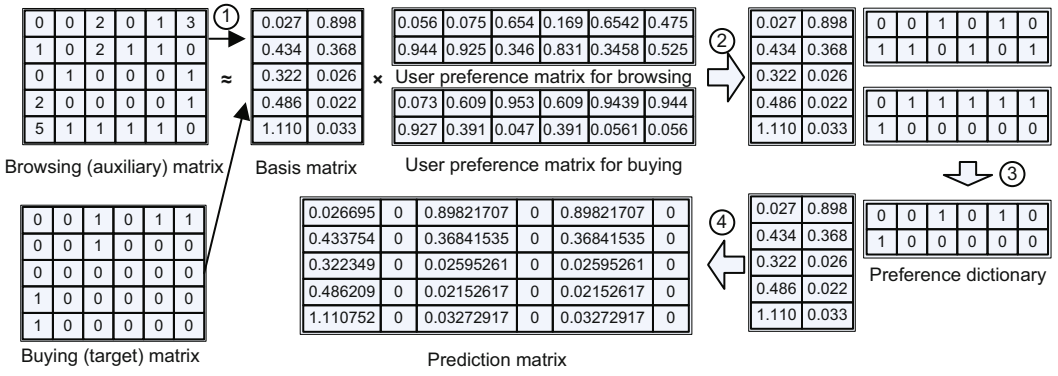

Fig. 3 Simulation of the preference transfer model for predicting user preference. By using the latent factor-user matrix, the simulation explains how to construct the consistent preference dictionary

be bought in the next time period. If the threshold is set at 0.4, P_{21} and P_{41} are also picked up. It is interesting that element P_{21} is selected. Although item 2 was not bought by user 1, it was visited in the current time. Furthermore, we can see the popularity of item 2 and activity of user 1 from the browsing matrix and the buying matrix. Thus, item 2 is considered to have greater possibility of being bought by user 1 in the future.

5.2 Real datasets: Tmall data

The Alibaba Group launched a recommendation algorithm competition based on Alibaba Tmall data designed to motivate researchers to improve the accuracy of the recommender system in the Tmall web site (Tmall Recommendation Prize 2014 of Alibaba Group, http://tianchi.aliyun.com/competition/index.htm?pageIndex=2). Alibaba provides four-month consumption records in the Tmall web site, including the browsing history and the purchasing behavior, to aid participants in predicting what brands of goods the consumer will buy in the following month. There are a total of 182 880 records related to 884 users and 9531 types of branded goods. Because the records of the buying and browsing behaviors for the fifth month are not available, we take the records of the buying matrix of the first three months as the target matrix in the training set, and the records of the browsing matrix for the last week in the third month as the auxiliary matrix. Likewise, we take the records of the first week in the fourth month as the test set.

5.3 Evaluation metrics

The prediction performances of all of the algorithms are measured by their precision and recall scores. The precision, P, of an algorithm is the fraction of the recommended set that is correct and is defined as follows:

$$P=\frac{\sum_{i=1}^{L}\text{hitItems}_i}{\sum_{i=1}^{L}\text{pItems}_i}, \tag{19}$$

where L is the number of all users who are in the recommended set, pItems_i the number of items recommended to user i, and hitItems_i the number of items that are actually selects from the recommended set. The recall, R, of an algorithm is the fraction of the correct set that is recommended and is defined as

$$R=\frac{\sum_{i=1}^{L}\text{hitItems}_i}{\sum_{i=1}^{B}\text{bItems}_i}, \tag{20}$$

where B is the number of all users in the correct set and bItems_i the number of items that user i actually selects from the correct set. If more items are recommended, the precision will decrease, but the recall will increase. The harmonic mean is called the F1 score, as shown in Eq. (21). The higher the F1 score is, the better the prediction performance is (Herlocker *et al.*, 2004).

$$\text{F1}=\frac{2PR}{P+R}. \tag{21}$$

5.4 Competing methods

Although we lack direct methods to transfer preference for implicit data, we choose the following similar methods to compare:

1. NMF: Without the auxiliary matrix, NMF is the baseline of single domain information for experiments. After matrix completion by multiplying the two factorized matrices, the top-n sorted elements in the reconstructive matrix $\boldsymbol{X}$ with values greater than 0.5 are recommended.

2. PMF: We use PMF as another kind of matrix factorization technique (different from non-negative factorization) to compare. After matrix completion by multiplying the two factorized matrices, the top-n sorted elements in the reconstructive matrix $\boldsymbol{X}$ with values greater than 0.5 are recommended.

3. JPP: Because the algorithm is designed primarily for document classification, we have to calculate $p(v_i|u_j)=\sum_z p(v_i|f_z)p(f_z|u_j)$ and select the top-n $p(v_i|u_j)$ for the prediction (v_i represents item i, u_j user j, and f_z latent factor z) (Cohn and Hofmann, 2000). The parameter of relative weight in this algorithm is set to 0.3 as suggested.

4. RMGM: In accordance with Li *et al.* (2009b), we modify the shared cluster-level rating model to the common cluster-level preferences model of the buying and browsing matrices. Because the user preference in the two domains can be viewed as the same structure, we set $K=L$. Finally, we use the user item bivariate probability as the prediction probability.

5. PTM: This is the proposed algorithm. In our experiments, we empirically set prior parameter

$a_u = 1.01$, $a_v = 1.01$, $a_w = 1.01$, $b_u = 0.1$, $b_v = 0.1$, $b_w = 0.1$, and threshold $\tau = 0.1$. The parameter study will be discussed later.

5.5 Parameter and hyperparameter setting

In our algorithm, there are three parameters: dimension size K, popularity θ, and threshold τ in the PTM. Fig. 6 shows that the predictive performance approaches an optimum when $K = 90$. As the dimension size K increases, overfitting occurs. Considering the most popular item's frequency to be single digit and a few popular items' frequencies to be more than 100, the popular θ is simply set to be the median value of the frequency set. The threshold τ controls the number of recommended items. The smaller the τ is, the larger the number of recommended items is. However, the F1 score does not have a direct linear relation with τ. We choose τ from 0.05 to 0.14 with a step size of 0.01. As shown in Fig. 4, the threshold is optimal at 0.07.

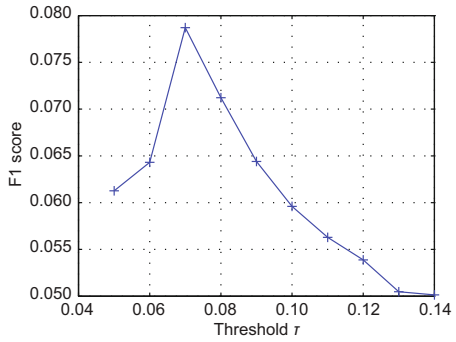

Fig. 4 F1 score influenced by different thresholds

The Gamma distribution $G(x; a, b)$ has the shape parameter a and the scale parameter b. To ensure that Eqs. (15)–(17) are non-negative, the shape parameter a must be set greater than one due to the occurrence of $a - 1$ in the numerator. Qualitatively, small values of a (for $a \to 1$) enforce sparse representations, and large values of $a > 2$ tie all values to be close to the nonzero mean (nonsparse representation) (Fig. 5). We hope the elements in the basis matrix $\boldsymbol{W}$ are close to nonzero mean and the elements in the coefficient matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ are sparse, so the hyperparameters are set empirically: $a_w = 3, b_w = 0.1, a_u = 1.01, b_u = 0.1, a_v = 1.01$, and $b_v = 0.1$.

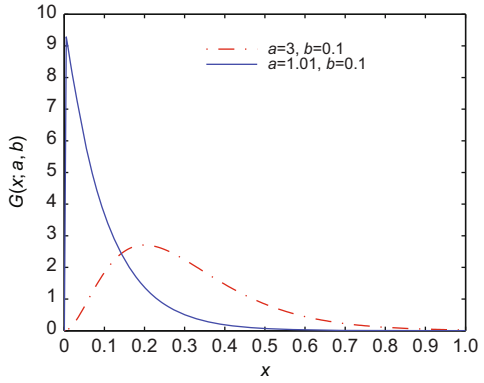

Fig. 5 Different shapes of Gamma distribution on different parameters

5.6 Results and analysis

1. Prediction performance: To reduce the effect of randomness, we repeat each trial 50 times using different methods mentioned above and compare these algorithms based on their average performances (Fig. 6). PMF and NMF are inferior to the cross-domain algorithms, such as JPP, RMGM, and PTM, due to no clue of purchase preference in the next period being considered. In contrast, PTM significantly outperforms the other algorithms mainly because it constructs the 'preference dictionary', which in advance captures the user preference of buying behavior hiding in the browsing behavior.

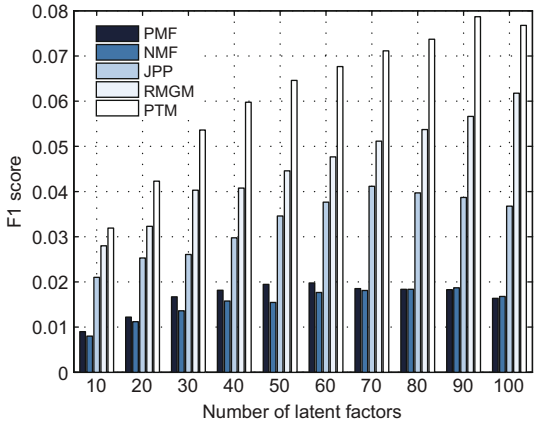

Fig. 6 Performance on collaboration prediction of all algorithms

2. Prediction effectiveness: We hypothesize that the browsing behavior before purchasing will benefit the prediction. To evaluate this hypothesis, we investigate the prediction of PTM by increasing the gap between the buying time in the testing set and the browsing time in the training sets (Table 1). The

Table 1 Prediction performance on four weeks

Algorithm	Precision			
	Week 1	Week 2	Week 3	Week 4
PMF	0.020 [0.018, 0.021]	0.020 [0.018, 0.021]	0.019 [0.016, 0.021]	0.019 [0.016, 0.021]
NMF	0.018 [0.013, 0.021]	0.018 [0.013, 0.020]	0.015 [0.012, 0.020]	0.014 [0.012, 0.017]
JPP	0.037 [0.035, 0.039]	0.035 [0.036, 0.037]	0.036 [0.035, 0.037]	0.033 [0.031, 0.035]
RMGM	0.058 [0.052, 0.063]	**0.055** [0.052, 0.062]	**0.041** [0.038, 0.044]	**0.041** [0.038, 0.042]
PTM	**0.078** [0.076, 0.080]	0.034 [0.032, 0.370]	0.021 [0.016, 0.022]	0.014 [0.011, 0.015]

[*]: 95% confidence intervals of performance

performance drops by 50% when the time gap increases to two weeks. Furthermore, the performance drops to the level which is similar to that of NMF when the gap increases to four weeks, which implies that there is no information transferred from the browsing interest to the buying interest. Taken together, the results show that PTM is sensitive to the time intervals between browsing and buying.

3. Algorithm convergence: We investigate the convergence of our PTM. Fig. 7 shows the convergence curve of PTM. The values of the cost function $\mathcal{L}$ with dimension size K being 10, 30, and 50 are plotted. As shown, the nonincreasing nature of $\mathcal{L}$ is obvious, and the loss function value drops very fast after a few iterations. However, the PTM model cannot ensure that the global minimum of the cost function can be obtained. Therefore, there may be multiple local minima, depending on the initial points. Nevertheless, our numerical experiments show that different initial values generate very similar results, which implies that the initial value might have only a small impact on the performance of the algorithm.

In terms of computing time, we compare our method with others on the same computer with the configuration of 2.13 GHz Intel® Xeon® CPU (i3330M) and 2.0 GB RAM. All these models are implemented using Matlab 2011a, and running times are listed in Table 2. PTM is better than RMGM but worse than others, which is mainly due to the larger costs on matrix computation in each iteration whilst solving the optimization problems.

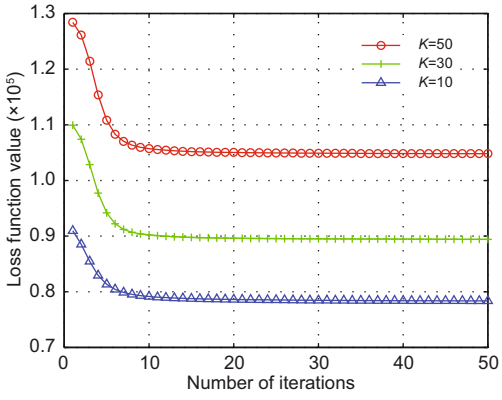

Fig. 7 The nonincreasing nature of the cost function for Alibaba Tmall data

Table 2 Time costs on Alibaba Tmall data with different algorithms

Algorithm	Time (s)	Algorithm	Time (s)
PMF	34.35	RMGM	363.19
NMF	20.88	PTM	213.75
JPP	183.12		

6 Conclusions and future work

In this study, we propose a novel method called the preference transfer model (PTM) for collaborative filtering with implicit data, especially for the extremely sparse implicit data. Different from existing analogous models based on Gaussian prior, our model is based on Poisson prior for multi-task nonnegative matrix factorization, which can capture the transition of users' preferences from browsing behavior to buying behavior. It incorporates homogeneous but cross-time-domain data sources. The experimental results demonstrate that the proposed PTM method outperforms the compared methods on Alibaba Tmall data.

Three parameters need to be learned, and they can be determined by performing grid searching on the training dataset. We found that PTM is sensitive to the time intervals between browsing time and buying time due to simply using element-wise multiplication of finding the consistent preference. In future work, we will investigate how to statistically quantify the relatedness between user-factor matrices in different domains. Furthermore, the preference transfer

model can be applied to other problems, such as predicting the frequency of users visiting sites and the frequency of users visiting advertisements, which will be another focus for us in the future.

References

Blei, D.M., Lafferty, J.D., 2006. Dynamic topic models. Int. Conf. on Machine Learning, p.113-120. http://dx.doi.org/10.1145/1143844.1143859

Chen, G., Wang, F., Zhang, C., 2009. Collaborative filtering using orthogonal nonnegative matrix tri-factorization. *Inform. Process. Manag.*, **45**(3):368-379. http://dx.doi.org/10.1016/j.ipm.2008.12.004

Cohn, D., Hofmann, T., 2000. The missing link—a probabilistic model of document content and hypertext connectivity. Conf. on Neural Information Processing Systems, p.430-436.

Devarajan, K., Wang, G., Ebrahimi, N., 2015. A unified statistical approach to non-negative matrix factorization and probabilistic latent semantic indexing. *Mach. Learn.*, **99**(1):137-163. http://dx.doi.org/10.1007/s10994-014-5470-z

Ding, C., Li, T., Peng, W., 2006. Nonnegative matrix factorization and probabilistic latent semantic indexing: equivalence, chi-square statistic, and a hybrid method. National Conf. on Artificial Intelligencer, p.342-347.

Gaussier, E., Goutte, C., 2005. Relation between PLSA and NMF and implications. Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, p.601-602. http://dx.doi.org/10.1145/1076034.1076148

Gopalan, P., Hofman, J.M., Blei, D.M., 2013. Scalable recommendation with Poisson factorization. arXiv:1311.1704. http://arxiv.org/abs/1311.1704

Gu, Q., Zhou, J., Ding, C., 2010. Collaborative filtering: weighted nonnegative matrix factorization incorporating user and item graphs. SIAM Int. Conf. on Data Mining, p.199-210. http://dx.doi.org/10.1137/1.9781611972801.18

Herlocker, J.L., Konstan, J.A., Terveen, L.G., *et al.*, 2004. Evaluating collaborative filtering recommender systems. *ACM Trans. Inform. Syst.*, **22**(1):5-53. http://dx.doi.org/10.1145/963770.963772

Hofmann, T., 2004. Latent semantic models for collaborative filtering. *ACM Trans. Inform. Syst.*, **22**(1):89-115. http://dx.doi.org/10.1145/963770.963774

Hofmann, T., Puzicha, J., 1999. Latent class models for collaborative filtering. Int. Joint Conf. on Artificial Intelligence, p.688-693.

Ju, B., Qian, Y., Ye, M., *et al.*, 2015. Using dynamic multitask non-negative matrix factorization to detect the evolution of user preferences in collaborative filtering. *PloS One*, **10**(8):e0135090. http://dx.doi.org/10.1371/journal.pone.0138279

Koren, Y., 2010. Collaborative filtering with temporal dynamics. *Commun. ACM*, **53**(4):89-97. http://dx.doi.org/10.1145/1721654.1721677

Koren, Y., Bell, R., Volinsky, C., 2009. Matrix factorization techniques for recommender systems. *Computer*, **42**(8):30-37. http://dx.doi.org/10.1109/MC.2009.263

Lathia, N., Hailes, S., Capra, L., 2009. Temporal collaborative filtering with adaptive neighbourhoods. Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, p.796-797. http://dx.doi.org/10.1145/1571941.1572133

Lee, D.D., Seung, H.S., 1999. Learning the parts of objects by non-negative matrix factorization. *Nature*, **401**(6755):788-791. http://dx.doi.org/10.1038/44565

Lee, D.D., Seung, H.S., 2000. Algorithms for non-negative matrix factorization. Conf. on Neural Information Processing Systems, p.556-562.

Li, B., Yang, Q., Xue, X., 2009a. Can movies and books collaborate? Cross-domain collaborative filtering for sparsity reduction. Int. Joint Conf. on Artificial Intelligence, p.2052-2057.

Li, B., Yang, Q., Xue, X., 2009b. Transfer learning for collaborative filtering via a rating-matrix generative model. Int. Conf. on Machine Learning, p.617-624. http://dx.doi.org/10.1145/1553374.1553454

Li, T., Sindhwani, V., Ding, C., *et al.*, 2010. Bridging domains with words: opinion analysis with matrix tri-factorizations. SIAM Int. Conf. on Data Mining, p.293-302. http://dx.doi.org/10.1137/1.9781611972801.26

Liu, J., Wang, C., Gao, J., *et al.*, 2013. Multi-view clustering via joint nonnegative matrix factorization. SIAM Int. Conf. on Data Mining, p.252-260. http://dx.doi.org/10.1137/1.9781611972832.28

Ma, H., Liu, C., King, I., *et al.*, 2011. Probabilistic factor models for web site recommendation. Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, p.265-274. http://dx.doi.org/10.1145/2009916.2009955

Mnih, A., Salakhutdinov, R., 2007. Probabilistic matrix factorization. Conf. on Neural Information Processing Systems, p.1257-1264.

Rendle, S., Freudenthaler, C., 2014. Improving pairwise learning for item recommendation from implicit feedback. ACM Int. Conf. on Web Search and Data Mining, p.273-282. http://dx.doi.org/10.1145/2556195.2556248

Rendle, S., Freudenthaler, C., Gantner, Z., *et al.*, 2009. BPR: Bayesian personalized ranking from implicit feedback. 25th Conf. on Uncertainty in Artificial Intelligence, p.452-461.

Salakhutdinov, R., Mnih, A., 2008. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. Int. Conf. on Machine Learning, p.880-887. http://dx.doi.org/10.1145/1390156.1390267

Savia, E., Puolamäki, K., Kaski, S., 2009. Latent grouping models for user preference prediction. *Mach. Learn.*, **74**(1):75-109. http://dx.doi.org/10.1007/s10994-008-5081-7

Shi, Y., Karatzoglou, A., Baltrunas, L., *et al.*, 2012. TFMAP: optimizing MAP for top-n context-aware recommendation. Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, p.155-164. http://dx.doi.org/10.1145/2348283.2348308

Si, L., Jin, R., 2003. Flexible mixture model for collaborative filtering. Int. Conf. on Machine Learning, p.704-711.

Singh, A.P., Gordon, G.J., 2008. Relational learning via collective matrix factorization. 14th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, p.650-658. http://dx.doi.org/10.1145/1401890.1401969

Su, X., Khoshgoftaar, T.M., 2009. A survey of collaborative filtering techniques. *Adv. Artif. Intell.*, **2009**:421425. http://dx.doi.org/10.1155/2009/421425

Xie, S., Lu, H., He, Y., 2012. Multi-task co-clustering via nonnegative matrix factorization. Int. Conf. on Pattern Recognition, p.2954-2958.

Zhang, S., Wang, W., Ford, J., *et al.*, 2006. Learning from incomplete ratings using non-negative matrix factorization. SIAM Int. Conf. on Data Mining, p.549-553. http://dx.doi.org/10.1137/1.9781611972764.58

Zhuang, F., Luo, P., Xiong, H., *et al.*, 2011. Exploiting associations between word clusters and document classes for cross-domain text categorization. *Stat. Anal. Data Min.: ASA Data Sci. J.*, **4**(1):100-114. http://dx.doi.org/10.1002/sam.10099

Appendix: Proof of the convergence of Eqs. (15)–(17)

Here we prove the convergence of the multiplicative updates of the preference transfer model (PTM). According to Lee and Seung (2000), we need to construct an auxiliary function $\mathcal{Q}$ of the loss function $\mathcal{L}$ on the conditions of satisfaction, i.e., $\mathcal{Q}(\boldsymbol{u},\boldsymbol{u})=\mathcal{L}(\boldsymbol{u})$ and $\mathcal{Q}(\boldsymbol{u},\boldsymbol{u}^t)\geq\mathcal{L}(\boldsymbol{u})$. Here, $\boldsymbol{u}$ is one column of matrix $\boldsymbol{U}$, $\mathcal{L}(\boldsymbol{u})$ the loss function of $\boldsymbol{u}$, and $\boldsymbol{u}^t$ the vector variable relative to $\boldsymbol{u}$ in auxiliary function $\mathcal{L}(\boldsymbol{u})$. Obviously, if the loss function $\mathcal{L}$ converges on the condition of multiplicative updates about $\boldsymbol{u}$, then it converges on the condition of MU about matrix $\boldsymbol{U}$.

Lemma 1

$$\begin{aligned}\mathcal{Q}(\boldsymbol{u},\boldsymbol{u}^t)=&\sum_{ia}W_{ia}u_a-\sum_{ia}x_i\frac{W_{ia}u_a^t}{\sum_b W_{ib}u_b^t}\left(\log W_{ia}u_a-\log\frac{W_{ia}u_a^t}{\sum_b W_{ib}u_b^t}\right)\\&+\sum_a\frac{u_a}{b_u}-(a_u-1)\sum_a\frac{u_a^t}{\sum_b u_b^t}\left(\log u_a-\log\frac{u_a^t}{\sum_b u_b^t}\right)\end{aligned}\tag{A1}$$

is an auxiliary function for $\mathcal{L}(\boldsymbol{u})=\sum_i\left(\sum_a W_{ia}u_a-x_i\log\sum_a(W_{ia}u_a)\right)+\sum_a u_a/b_u-(a_u-1)\log\sum_a u_a$.

Proof It is straightforward to verify

$$\begin{aligned}\mathcal{Q}(\boldsymbol{u},\boldsymbol{u})=&\sum_{ia}W_{ia}u_a-\sum_{ia}x_i\frac{W_{ia}u_a}{\sum_b W_{ib}u_b}\log\sum_a W_{ia}u_a+\sum_a\frac{u_a}{b_u}-(a_u-1)\sum_a\frac{u_a}{\sum_b u_b}\log\sum_a u_a\\=&\sum_{ia}W_{ia}u_a-\sum_{ia}x_i\log\sum_a W_{ia}u_a+\sum_a\frac{u_a}{b_u}-(a_u-1)\log\sum_a u_a\\=&\mathcal{L}(\boldsymbol{u}).\end{aligned}\tag{A2}$$

Proving $\mathcal{Q}(\boldsymbol{u},\boldsymbol{u}^t)\geq\mathcal{L}(\boldsymbol{u})$ is equivalent to proving

$$\begin{aligned}\mathcal{Q}(\boldsymbol{u},\boldsymbol{u}^t)-\mathcal{L}(\boldsymbol{u})=&\sum_i\left(x_i\log\sum_a(W_{ia}u_a)-x_i\frac{W_{ia}u_a^t}{\sum_b W_{ib}u_b^t}\left(\log W_{ia}u_a-\log\frac{W_{ia}u_a^t}{\sum_b W_{ib}u_b^t}\right)\right)\\&+(a_u-1)\left(\log\sum_a u_a-\frac{u_a^t}{\sum_b u_b^t}\left(\log u_a-\log\frac{u_a^t}{\sum_b u_b^t}\right)\right)\geq 0.\end{aligned}\tag{A3}$$

Because $x_i\geq 0$ and $a_u>1$, proving inequality (A3) is equal to proving

$$\log\sum_a(W_{ia}u_a)\geq\sum_a\frac{W_{ia}u_a^t}{\sum_b W_{ib}u_b^t}\left(\log W_{ia}u_a-\log\frac{W_{ia}u_a^t}{\sum_b W_{ib}u_b^t}\right)\tag{A4}$$

and

$$\log\sum_a u_a\geq\sum_a\frac{u_a^t}{\sum_b u_b^t}\left(\log u_a-\log\frac{u_a^t}{\sum_b u_b^t}\right).\tag{A5}$$

We use the convexity of the logarithm function to drive the two inequalities

$$\log\sum_a (W_{ia}u_a) \geq \sum_a \alpha_a \log\frac{W_{ia}u_a}{\alpha_a}, \quad \log\sum_a u_a \geq \sum_a \beta_a \log\frac{u_a}{\beta_a}, \tag{A6}$$

which hold for all non-negative α_a that sum to unity, as well as β_a. Setting

$$\alpha_a = \frac{W_{ia}u_a^t}{\sum_b W_{ib}u_b^t}, \quad \beta_a = \frac{u_a^t}{\sum_b u_b^t}, \tag{A7}$$

we obtain inequalities (A4) and (A5). Thus, $\mathcal{Q}(\boldsymbol{u},\boldsymbol{u}^t) \geq \mathcal{L}(\boldsymbol{u})$ holds.

According to the lemma proved in Lee and Seung (2000), if $\mathcal{Q}$ is an auxiliary function, then $\mathcal{L}$ is nonincreasing under the update

$$\boldsymbol{u}^{t+1} = \arg\min_{\boldsymbol{u}} \mathcal{Q}(\boldsymbol{u},\boldsymbol{u}^t). \tag{A8}$$

The minimum of $\mathcal{Q}(\boldsymbol{u},\boldsymbol{u}^t)$ with respect to $\boldsymbol{u}$ is determined by setting the gradient to zero:

$$\frac{\partial\mathcal{Q}(\boldsymbol{u},\boldsymbol{u}^t)}{\partial u_a} = \sum_i W_{ia} + \frac{1}{b_u} - \sum_i x_i \frac{W_{ia}u_a^t}{\sum_b W_{ib}u_b^t}\frac{1}{u_a} - (a_u-1)\frac{u_a^t}{\sum_b u_b^t}\frac{1}{u_a} = 0. \tag{A9}$$

Thus, the update rule of $\boldsymbol{u}$ takes the form

$$u_a^{t+1} = u_a^t \frac{\sum\limits_i W_{ia}\dfrac{x_i}{\sum_b W_{ib}u_b^t} + \dfrac{a_u-1}{\sum_b u_b^t}}{\sum\limits_i W_{ia} + \dfrac{1}{b_u}}. \tag{A10}$$

Rewritten in matrix form, this is equivalent to the update rule as follows:

$$U_{kj} \leftarrow U_{kj} \frac{\sum_i \dfrac{W_{ik}X_{ij}}{\sum_{k=1}^K W_{ik}U_{kj}} + \dfrac{a_u-1}{U_{kj}}}{\sum_i W_{ik} + \dfrac{1}{b_u}}. \tag{A11}$$

Similarly, it can be proved that the update rules for $\boldsymbol{W}$ and $\boldsymbol{V}$ are nonincreasing.