Personalized Recommendation via Cross-Domain Triadic Factorization

Liang Hu
Department of CSE
Shanghai Jiaotong University
lianghu@sjtu.edu.cn

Jian Cao*
Department of CSE
Shanghai Jiaotong University
cao-jian@sjtu.edu.cn

Guandong Xu
Advanced Analytics Institute
University Technology Sydney
guandong.xu@uts.edu.au

Longbing Cao
Advanced Analytics Institute
University Technology Sydney
longbing.cao@uts.edu.au

Zhiping Gu
Department of Electrical Engineering
Shanghai Technical Institute of
Electronics & Information
guzhiping@stiei.edu.cn

Can Zhu
Department of CSE
Shanghai Jiaotong University
0627wshhg@sjtu.edu.cn

ABSTRACT

Collaborative filtering (CF) is a major technique in recommender systems to help users find their potentially desired items. Since the data sparsity problem is quite commonly encountered in real-world scenarios, Cross-Domain Collaborative Filtering (CDCF) hence is becoming an emerging research topic in recent years. However, due to the lack of sufficient dense explicit feedbacks and even no feedback available in users' uninvolved domains, current CDCF approaches may not perform satisfactorily in user preference prediction. In this paper, we propose a generalized Cross Domain Triadic Factorization (CDTF) model over the triadic relation *user-item-domain*, which can better capture the interactions between domain-specific user factors and item factors. In particular, we devise two CDTF algorithms to leverage user explicit and implicit feedbacks respectively, along with a genetic algorithm based weight parameters tuning algorithm to trade off influence among domains optimally. Finally, we conduct experiments to evaluate our models and compare with other state-of-the-art models by using two real world datasets. The results show the superiority of our models against other comparative models.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Filtering; I.2.6 [**Artificial Intelligence**]: Parameter Learning

General Terms

Algorithms, Performance, Experimentation, Human Factors

Keywords

Recommender System, Cross-Domain Collaborative Filtering, Triadic Factorization

1. INTRODUCTION

ACM 978-1-4503-2035-1 /13/05.The huge and ever fast increasing amount of information on the Internet has penetrated every corner of our life. However, we become more easily overwhelmed by so much information and unable to find what we really desire. When we follow events on Facebook, buy books on Amazon or add apps into a smartphone, systems may record our feedbacks, e.g., a rating assigned to a book. Based on such observed feedbacks (or ratings) collected from like-minded users, collaborative filtering (CF) in recommender systems can predict personalized preferences to unconsumed items. In general, CF methods can be sub-divided into neighborhood-based and model-based approaches [5; 22; 26]. Therein, latent factor model based on matrix factorization (MF) [6; 9] has gained the dominance in recent years.

The essence to success in CF is highly dependent on the feedback data. However, users are not always willing to provide feedbacks due to various personal reasons. Even some applications possess the data sparsity problem in nature, for instance, users who has bought a new car recently may not have a new car purchase plan in next five years. Thus most CF methods, including MF, suffer from the data sparsity [26] and cold-start [9; 23] issues. The lack of reliable feedback data has become a major barrier for CF methods.

To deal with the sparsity issue, Cross-Domain Collaborative Filtering (CDCF), which leverages the information from multiple related domains, is an emerging research topic in recent years. Some CDCF algorithms have been proposed in literatures [11], where the basic idea is based on the assumption of the existence of multiple related domains and the user preference learned from one dense domain, e.g. movies, can be re-used to make prediction in a sparse domain, e.g. books (i.e. cross domain learning) [12; 18]. An early neighborhood based CDCF (N-CDCF) was mentioned in [1], but it can only provide a very local optimum solution as done by neighborhood based CF models (further analysis provided in the next section). Recently, some cross-domain matrix factorization (CDMF) models [18; 24] have been proposed to overcome the local optimum problem of N-CDCF. The underlying idea of CDMF can be illustrated using Figure 1 (b), where user factor matrix **U** serves as the bridge to transfer knowledge from auxiliary domain (A) to target domain (T).

Most CDMF models assume the auxiliary data is relative dense for all users or items [18]. However, we argue that this assumption is not always true in real world. In general, our argument is based on the well-known power law, as illustrated in Figure 1(a), only the minority of users are rating frequently while the majority of users are quite inactive in providing feedback. This observation might impact the hypothesis of traditional CDCF approaches, therefore resulting in the deterioration of recommendation performance.

*Jian Cao is the corresponding author

WWW 2013, May 13--17, 2013, Rio de Janiero, Brazil.
ACM 978-1-4503-2035-1 /13/05.

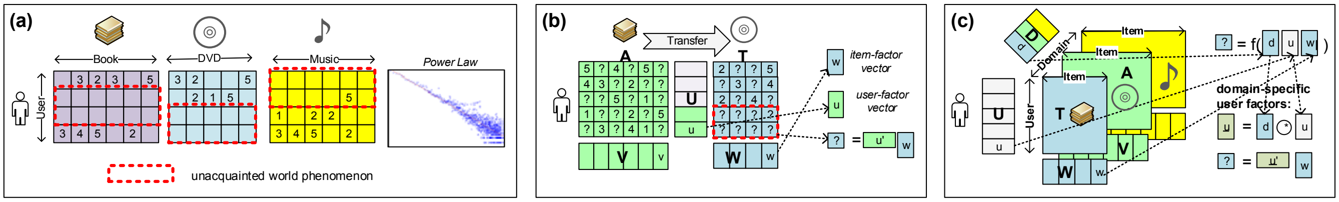


Figure 1: (a) Due to the power law, the feedbacks over majority of users are sparse in each domain, so the unacquainted-world phenomenon are ubiquitous in CDCF; (b) The demonstration of unacquainted-world issue in CDMF; (c) More accurate triadic factor analysis over CDCF.

Furthermore, due to the diversity of user interests a user is usually active in a few domains that she/he is really interested in, but silent in other domains hardly involved. Given a set of domains, we call those user's uninvolved domains as ***unacquainted world***. Since each user has different domains of interest, the unacquainted-world phenomenon is common in CDCF problem, shown in Figure 1 (a). Moreover, the ubiquitous unacquainted-world phenomenon may negatively impact the recommendation performance of CDMF models in heterogeneous domain settings. Consider the example depicted in Figure 1 (b), CDMF aims to improve recommendation in the target domain *T* by utilizing the transferred knowledge (i.e. the user factor matrix **U**) from the auxiliary domain *A*. More specifically, this transferred user factor matrix **U** should be updated by taking into account users' feedbacks from *T* before serving as the user factor matrix **U** for target domain *T* [18; 24]. Now the problem occurs: no feedback is available for the last two users in *T* (i.e. the unacquainted world, marked with dashed red box) to adjust the transferred user-factor vector **u**, but the prediction of preference to an item in *T* is purely determined by the operation of $\mathbf{u}^{\mathrm{T}}\mathbf{w}$, which may yield an inaccurate result due to the unadjusted **u** and the heterogeneity of item factors between the heterogeneous domains *A* and *T* (e.g. the heterogeneity between **v** and **w**). Therefore, it results in unreliable prediction completely based on the preference learned from a heterogeneous domain. Thus this raises a demand to devise a new cross-domain learning model by jointly leveraging the complementary data from multiple domains rather than simply relying on some dense feedback domain.

The major reason caused the above concerns is that CDMF deals with a set of *user-item* data over multiple domains in a flat manner but it does not consider the attribute of domain factor. The absence of domain-specific information in factorization process leads CDMF to suffering from the unacquainted-world issue. We argue that domain factors is an essential element in cross "domain" problem, so cross domain learning should take into consideration the full triadic relation *user-item-domain* to reveal the user preference on items within various domains in depth, rather than the dyadic relation *user-item* modeled by CDMF.

To learn such triadic factors from data, intuitively a tensor factorization (TF) could be introduced. However, a standard TF model requires that the slice of each domain should be the same size. Obviously, overlying all domain slices (differ in the number of items) in Figure 1 (c) cannot form a cubical tensor. Inspired by PARAFAC2 [7], a special TF model, we propose Cross-Domain Triadic Factorization (CDTF) to relax the constraint that the same item-factor matrix is employed for all domains. As illustrated in Figure 1 (c), CDTF allows an exclusive item-factor matrix for each domain to express heterogeneities. In addition, user-factor matrix **U** in CDTF is used to model the general users concerns over all domains and the domain-factor matrix **D** carries the information to express the traits of each domain. Hence each observation can be viewed as the result of triadic interaction among user, item and domain factors. Further, we can interpret that the domain-specific user factors is generated by the interaction between domain factors and general user factors as shown in Figure 1 (c). Obviously, such triadic factor model avoids the unacquainted-world issue in CDMF.

In real-world scenarios, another difficulty is that the user explicit feedback data (e.g., ratings) are sometimes hardly available. How to alleviate this kind of problems becomes a new research direction in CF and more and more studies attempt to make use of implicit feedbacks (here implicit feedback means the intention conveyed by user activities, such as purchase history or browsing behavior) [6; 20]. Accordingly, we further extend CDTF model to accommodate implicit feedbacks, namely CDTF-IF, which can effectively deal with the one-class implicit feedback data that CDTF cannot handle. Moreover, in cross domain learning problems, tuning the trade-off parameters over domains is an essential step to achieve better performance [16; 24]. Therefore, in this work we also investigate an automated and robust trade-off parameters determination approach for our models based on genetic algorithm (GA).

The contributions of this paper can be summarized as follows:

- We address the CDCF problem by formulating a generalized triadic relation *user-item-domain*.
- We devise a cross-domain triadic factorization (CDTF) model to learn the triadic factors for user, item and domain, where the item dimensionality varies with domains.
- To alleviate the absence of explicit feedbacks, we extend our proposed CDTF model to be able to utilize the implicit feedbacks that CDTF cannot handle.
- We study an automated optimal weight parameter estimation algorithm based on genetic algorithm.
- We perform experiments on two real world datasets to evaluate our models and make comparisons with other state-of-the-art models.

2. CDCF FROM CLASSICAL CF VIEWS

Neighborhood and MF based methods are two kinds of dominant approaches in CF. Although such classical CF methods can be applied to CDCF, they have disadvantages in nature.

Notations: $\boldsymbol{D} = \{D_1, D_2, \cdots, D_K\}$ denotes all the domains for modeling, $\boldsymbol{U} = \{u_1, u_2, \cdots, u_N\}$ denotes the users in $\boldsymbol{D}$ and $\boldsymbol{P}_{D_k} = \{p_1^{D_k}, p_2^{D_k}, \cdots, p_M^{D_k}\}$ denotes items belonging to the domain D_k.

N-CDCF: Neighborhood based CF compute similarity between users or items, which can be sub-divided into two types: user-based nearest neighbor and item-based nearest neighbor [26].

For a user-based CDCF algorithm, we first calculate the similarity, $w_{u,v}$, between the users u and v who have co-rated the same set of items. The similarity can be measured by the Pearson correlation:

$$w_{u,v} = \frac{\sum_{p \in \boldsymbol{p}_{u,v}} (r_{u,p} - \bar{r}_u)(r_{v,p} - \bar{r}_v)}{\sqrt{\sum_{p \in \boldsymbol{p}_{u,v}} (r_{u,p} - \bar{r}_u)^2 \sum_{p \in \boldsymbol{p}_{u,v}} (r_{v,p} - \bar{r}_v)^2}} \tag{1}$$

where $\boldsymbol{p}_{u,v} = \boldsymbol{p}_u \cap \boldsymbol{p}_v$ ($\boldsymbol{p}_u = \bigcup_{d \in \boldsymbol{D}} \boldsymbol{p}_u^d$, $\boldsymbol{p}_v = \bigcup_{d \in \boldsymbol{D}} \boldsymbol{p}_v^d$) denotes the items over all domains $\boldsymbol{D}$ co-rated by u and v; $r_{u,p}$ and $r_{v,p}$ are the ratings on item p given by users u and v respectively; $\bar{r}_u$ is the average rating of user u for all the items rated. Then, the predicted rating of an item p for user u can be calculated by a weighted average strategy [22]:

$$\hat{r}_{u,p} = \bar{r}_u + \frac{\sum_{v \in \boldsymbol{U}_{u,p}^k} w_{u,v} (r_{v,p} - \bar{r}_v)}{\sum_{v \in \boldsymbol{U}_{u,p}^k} |w_{u,v}|} \tag{2}$$

where $\boldsymbol{U}_{u,p}^{\boldsymbol{k}}$ denotes the set of top k users (k neighbors) that are most similar to user u who rated item p.

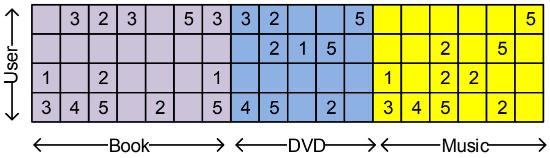
Similar to user-based algorithm, the item-based CDCF needs to compute the similarity, $w_{p,q}$, between item pair p and q. Given co-rated cases $U_{p,q}$ over p and q, i.e. each case is that a user rated both p and q, the Pearson correlation is given by:

$$w_{p,q} = \frac{\sum_{u \in U_{p,q}} (r_{u,p} - \bar{r}_p)(r_{u,q} - \bar{r}_q)}{\sqrt{\sum_{u \in U_{p,q}} (r_{u,p} - \bar{r}_p)^2 \sum_{u \in U_{p,q}} (r_{u,q} - \bar{r}_q)^2}} \tag{3}$$

Then, the predicted value, $\hat{r}_{u,p}$, is taken as a weighted average of the ratings for neighboring k items rated by u, denoted $\boldsymbol{P}_{u,p}^k$.

$$\hat{r}_{u,p} = \bar{r}_p + \frac{\sum_{q \in \boldsymbol{P}_{u,p}^k :} w_{p,q} (r_{u,q} - \bar{r}_q)}{\sum_{v \in \boldsymbol{U}_{u,p}^k} |w_{p,q}|} \tag{4}$$

MF-CDCF: The method to perform MF on a CDCF problem is straightforward. We can construct a matrix $\boldsymbol{M}_{D_k}$ that takes all users $\boldsymbol{U}$ as the rows and all items $\boldsymbol{P}_{D_k}$ in domain D_k as the columns. Thus, we easily obtain K matrices $\boldsymbol{M}_{D_1}, \boldsymbol{M}_{D_2}, \cdots, \boldsymbol{M}_{D_K}$ for K domains. Then, an augmented matrix, $\boldsymbol{M}_{\boldsymbol{D}}$, can be built by horizontally concatenating all matrices as shown in Figure 2.



	3	2	3		5	3	3	2			5						5
								2	1	5				2		5	
1		2				1						1		2	2		
3	4	5		2		5	4	5		2		3	4	5		2	

Figure 2: Horizontal concatenation of matrices for all domains

With the matrix $\boldsymbol{M}_{\boldsymbol{D}}$ in hand, we can exploit any classical MF algorithm, e.g. the frequently used stochastic gradient descent (SGD) method [9], to construct user factor matrix and item factor matrix. These factor matrices are used for prediction.

Disadvantage: Neighborhood based models are most effective at detecting much localized relationships and unable to capture the totality of weak signals encompassed in all of a user's ratings. For example, u_1 rated items $\{p_1, p_2\}$, u_2 rated items $\{p_3, p_4\}$ and u_3 rated items $\{p_2, p_3\}$. The direct correlation between u_1 and u_2 is zero. In fact, u_1 is correlated with u_3 by p_2 and u_2 is correlated with u_3 by p_3, so u_1 is transitively correlated with u_2 instead of zeros. It proves that N-CDCF cannot obtain a global optimal solution, especially when the data is very sparse.

MF-CDCF accommodates items from all domains into a single matrix so as to employ single-domain MF. However, single domain model assumes the homogeneity of items. Obviously, item factors for different domains may quite heterogeneous so MF-CDCF fails to express them. Furthermore, such model absolutely loses the information to model domain factors for CDCF problem.

3. OUR MODELS

3.1 Preliminary

Before clarifying our model, we firstly introduce some basic notations, operations and algorithms for TF models. There are different TF models in literatures, such as Tucker model, CP model (canonical decomposition/parallel factor analysis (PARAFAC)) [8; 15]. Here, we mainly focus on CP model because our model is an extension of PARAFAC2 which needs to cope with CP.

3.1.1 Notations and Operations

The order of a tensor is the number of dimensions, also known as ways or modes. In this paper, tensors are denoted by boldface script letter, e.g. $\boldsymbol{\mathcal{X}}$. Matrices are denoted by boldface capital letters, e.g. $\mathbf{X}$. Vectors are denoted by boldface lowercase letters, e.g. $\mathbf{x}$. Entries are denoted lowercase letters with subscripts e.g. $x_{i,j,k}$. In addition, we denote the ith row of a matrix $\mathbf{X}$ as $\mathbf{X}_{i,\cdot}$, the jth column as $\mathbf{X}_{\cdot,j}$ and $\mathbf{X}_{i,j}$ for the entry (i,j).

The nth mode matricizing operation maps a tensor into a matrix, e.g., $\mathbf{X}_{(2)}$ represents the mapping $\boldsymbol{\mathcal{X}}^{I \times J \times K} \rightarrow \mathbf{X}_{(2)}^{J \times IK}$ [8; 15]. $\otimes$ denotes the Kronecker product and $\odot$ denotes the Khatri-Rao product, e.g. $\mathbf{X} \odot \mathbf{Y} = [\mathbf{X}_{\cdot,1} \otimes \mathbf{Y}_{\cdot,1}\ \mathbf{X}_{\cdot,2} \otimes \mathbf{Y}_{\cdot,2} \cdots \mathbf{X}_{\cdot,R} \otimes \mathbf{Y}_{\cdot,R}]$. $\circledast$ is the element-wise product while $\oslash$ is the element-wise division. $\langle \boldsymbol{\mathcal{X}} \circledast \boldsymbol{\mathcal{Y}} \rangle = \sum_{i,j,k} x_{i,j,k} y_{i,j,k}$ denotes the inner product and the norm of a tensor is, $\|\boldsymbol{\mathcal{X}}\| = \sqrt{\langle \boldsymbol{\mathcal{X}} \circledast \boldsymbol{\mathcal{X}} \rangle}$.

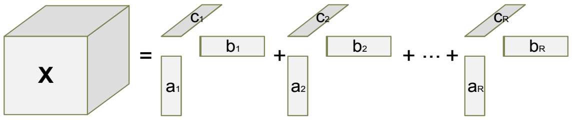


Figure 3: The CP factorization of a three-order tensor

3.1.2 CP Model

CP model decomposes a tensor into a sum of rank-one components as illustrated in Figure 3. For instance, given a three-order tensor $\boldsymbol{\mathcal{X}}$, the factorization can be written as $\boldsymbol{\mathcal{X}} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] = \sum_{r=1}^{R} \mathbf{A}_{\cdot,r} \circ \mathbf{B}_{\cdot,r} \circ \mathbf{C}_{\cdot,r}$, where $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are R-component factor matrices and $\circ$ denotes the outer product, i.e. the entries are computed $x_{i,j,k} = \sum_{r=1}^{R} \mathbf{A}_{i,r} \mathbf{B}_{j,r} \mathbf{C}_{k,r}$. Let $\boldsymbol{\mathcal{X}}$ be a three-order tensor with the size $I \times J \times K$. We can formulate the problem of fitting $\boldsymbol{\mathcal{X}}$ as a least squares optimization problem [8]:

$$\min f(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \frac{1}{2} \|\boldsymbol{\mathcal{X}} - [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]\|^2 + \frac{\lambda_A}{2} \|\mathbf{A}\|_F^2 + \frac{\lambda_B}{2} \|\mathbf{B}\|_F^2 + \frac{\lambda_C}{2} \|\mathbf{C}\|_F^2 \tag{5}$$

where regularization terms are added to avoid overfitting,$\|\cdot\|_F$ is the Frobenius norm and $\lambda_A, \lambda_B, \lambda_C$ are regularization parameters.

It is easy to prove that the partial derivative of the objective function (5) w.r.t. $\mathbf{A}$ is given by:

$$\frac{\partial f}{\partial \mathbf{A}} = (\mathbf{X}_{(1)} - \mathbf{Y}_{(1)})(\mathbf{C} \odot \mathbf{B}) + \lambda_A \mathbf{A}$$

where $\mathbf{Y} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$. Setting the above equation equal to zero and the property of pseudo-inverse of Khatri-Rao product [27] yields:

$$\mathbf{A} = \mathbf{X}_{(1)} (\mathbf{C} \odot \mathbf{B}) (\mathbf{B}^{\mathrm{T}} \mathbf{B} \circledast \mathbf{C}^{\mathrm{T}} \mathbf{C} + \lambda_A \mathbf{I})^{\dagger} \tag{6}$$

Similarly, the optimal solutions w.r.t. $\mathbf{B}$ and $\mathbf{C}$ are given by:

$$\mathbf{B} = \mathbf{X}_{(2)}(\mathbf{C} \odot \mathbf{A})(\mathbf{A}^{\mathrm{T}}\mathbf{A} \circledast \mathbf{C}^{\mathrm{T}}\mathbf{C} + \lambda_B\mathbf{I})^{\dagger} \qquad (7)$$

$$\mathbf{C} = \mathbf{X}_{(3)}(\mathbf{B} \odot \mathbf{A})(\mathbf{B}^{\mathrm{T}}\mathbf{B} \circledast \mathbf{A}^{\mathrm{T}}\mathbf{A} + \lambda_C\mathbf{I})^{\dagger} \qquad (8)$$

Above derivation corresponds to a regularized alternative least square algorithm, CP-ALS-R, given by Algorithm 1. The complexity for this algorithm is proportional to $IJKR + (I + J + K)R^2$, per iteration. Since we normally have $IJK \gg (I + J + K)$, the computational complexity is $\mathrm{O}(IJKR)$.

```
Algorithm 1: [$\mathbf{A},\mathbf{B},\mathbf{C}$] = CP-ALS-R($\mathcal{X}, \lambda_A, \lambda_B, \lambda_C$)
Input: $\mathcal{X}$ the tensor for factorization,
       $\lambda_A, \lambda_B, \lambda_C$ the regularization paramters
Output: $\mathbf{A},\mathbf{B},\mathbf{C}$ the factor matrices
Begin:
1:  Initialize $\mathbf{A},\mathbf{B},\mathbf{C}$
2:  Fix $\mathbf{B},\mathbf{C}$: Update $\mathbf{A}$ by Equation (6)
3:  Fix $\mathbf{A},\mathbf{C}$: Update $\mathbf{B}$ by Equation (7)
4:  Fix $\mathbf{A},\mathbf{B}$: Update $\mathbf{C}$ by Equation (8)
5:  Repeat 2 – 4 until convergence
6:  Return $\mathbf{A},\mathbf{B},\mathbf{C}$
End
```

3.2 Cross-Domain Triadic Factorization

We have discussed the weaknesses of traditional CDCF approaches in the previous section, where items from all domains are mixed together, so the item latent factors cannot be well learned due to the heterogeneity between domains. In addition, all those approaches discard the most important domain-specific information.

3.2.1 Model

A straightforward method to capture the 3-way interaction between *user-item-domain* is to model this triadic relation by a cube, i.e. a three order tensor, where each frontal slice in this cube corresponds to a rating matrix for each domain. Unfortunately, the inconsistent number of items for each domain, as illustrated in the left part of Figure 4, cannot form a standard tensor. PARAFAC2 [8; 15] relaxes CP's constraints that apply the same factors across a parallel set of matrices. Inspired by this idea, we propose the cross-domain triadic factorization (CDTF) model, which can be applied to a collection of rating matrices for domains that are equivalent in the *User* dimension but vary in the *Item* dimensions over domains.

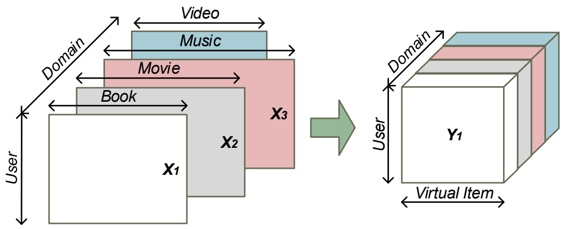


Figure 4: Slices of domain-specific matrices with heterogeneous items are transformed into a cubical tensor containing virtual items with identical length.

The standard CP model presented previously can be written as the factorization form w.r.t. each slices, $\mathbf{Y}_k$, for k=1 to K

$$\mathbf{Y}_k = \mathbf{A}\mathbf{\Sigma}_k\mathbf{B}^{\mathrm{T}} + \mathbf{E}_k \qquad (9)$$

where $\mathbf{A},\mathbf{B}$ are the factor matrices as given in previous section, $\mathbf{\Sigma}_k = \mathrm{diag}(\mathbf{C}_{k,:})$ is an $R \times R$ diagonal matrix of weights for the slice $\mathbf{Y}_k$, and $\mathbf{E}_k$ denotes the residuals [11; 27]. Therefore, we can rewrite the objective function (5) w.r.t. the slices $\mathbf{Y}_k$ for k=1 to K

$$\min f(\mathbf{A},\mathbf{B},\mathbf{C}) = \frac{1}{2}\sum_{k=1}^{K}\left\|\mathbf{Y}_k - \mathbf{A}\mathbf{\Sigma}_k\mathbf{B}^{\mathrm{T}}\right\|^2 + \frac{\lambda_A}{2}\|\mathbf{A}\|_F^2 + \frac{\lambda_B}{2}\|\mathbf{B}\|_F^2 + \frac{\lambda_C}{2}\|\mathbf{C}\|_F^2 \qquad (10)$$

Let us denote $\mathbf{X}_{\boldsymbol{D}} = \{\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_K\}$ to be the rating matrices for domains, where each matrix, $\mathbf{X}_k$, has the size $N \times M_k$, N is the number of users and M_k is the number of items in the kth domain. We apply a PARAFAC2-like modeling strategy to the collection of rating matrices, $\mathbf{X}_{\boldsymbol{D}}$, with varying sizes in *Item* mode (see Figure 4). Analogous to Eq. (9) for CP model, we can write the similar form of factorization w.r.t. the rating matrix of each domain.

$$\mathbf{X}_k = \mathbf{U}\mathbf{\Sigma}_k\mathbf{V}_k^{\mathrm{T}} + \mathbf{E}_k \qquad (11)$$

where $\mathbf{U}$ denotes the $N \times R$ factor matrix (it refers to user factors in our model), $\mathbf{V}_k$ is the $M_k \times R$ factor matrix (it refers to item factors in our model) for the slice $\mathbf{X}_k$ and $\mathbf{\Sigma}_k$ is an $R \times R$ diagonal matrix (it refers to domain factor in our model) for slice $\mathbf{X}_k$. Then, we easily obtain the objective function with the same form as Eq. (12). However, such PARAFAC2-like factorization is not unique without additional constraints [11; 27]. To improve the uniqueness property, Harshman [4] imposed a constraint that the cross product $\mathbf{V}_k^{\mathrm{T}}\mathbf{V}_k$ is a invariant matrix over k, i.e., $\mathbf{\Phi} = \mathbf{V}_k^{\mathrm{T}}\mathbf{V}_k$ for k=1, … , K. Thus, Eq. (11) can be written as:

$$\mathbf{X}_k = \mathbf{U}\mathbf{\Sigma}_k(\mathbf{P}_k\mathbf{V})^{\mathrm{T}} + \mathbf{E}_k \qquad (12)$$

where $\mathbf{U},\mathbf{\Sigma}_k$ are defined as usual, $\mathbf{P}_k$ is a column-wise orthonormal matrix (i.e. $\mathbf{P}_k^{\mathrm{T}}\mathbf{P}_k = \mathbf{I}$) of size $M_k \times R$ and $\mathbf{V}$ is an $R \times R$ matrix that does not vary by slice. The cross-product constraint is enforced implicitly since

$$\mathbf{V}_k^{\mathrm{T}}\mathbf{V}_k = (\mathbf{P}_k\mathbf{V})^{\mathrm{T}}(\mathbf{P}_k\mathbf{V}) = \mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{\Phi}$$

Then, the objective function can be given according to Eq. (12)

$$\min f(\mathbf{U},\mathbf{V},\mathbf{C},\mathbf{P}_k) = \frac{1}{2}\sum_{k=1}^{K}\left\|\mathbf{X}_k - \mathbf{U}\mathbf{\Sigma}_k(\mathbf{P}_k\mathbf{V})^{\mathrm{T}}\right\|^2 \qquad (13)$$

Weights over Slices: In our model, the user factor matrix $\mathbf{U}$ is shared across all domains (see Eq. (13)), i.e. learning $\mathbf{U}$ is affected by the loss on each slice. Some domain may have a lot of items and feedbacks (heavy slice) while other domain may only have a few of items and a few feedbacks (light slice). If the loss from a heavy slice overwhelms the loss from light slices, $\mathbf{U}$ is fully determined by the heavy slice. On the other hand, the scale of ratings on each slice may be different, e.g. the ratings on some slices are in the range of 1-5 and others may be 1-100, so the larger-scaled rating slice tends to account for more loss. More importantly, sometimes we deliberately require that the learning of $\mathbf{U}$ is mainly determined by feedbacks from target domain so as to perform better prediction.

Consequently, we add the weight parameter, w_k, to the objective function (13) to adjust the penalty of loss on each slice as given by Eq. (14). If we assign a large weight to some domain slice, then factor matrix $\mathbf{U}$ is mainly learned from the factorization over this slice. Note that the change of $\mathbf{U}$ will update other factor matrices $\mathbf{V},\mathbf{C},\mathbf{P}_k$ in turn during the process of factorization. Therefore, we can control the learning result of all factor matrices by tuning the weight assigned to each slice.

$$\min f(\mathbf{U},\mathbf{V},\mathbf{C},\mathbf{P}_k) = \frac{1}{2}\sum_{k=1}^{K}\left\|w_k(\mathbf{X}_k - \mathbf{U}\mathbf{\Sigma}_k(\mathbf{P}_k\mathbf{V})^{\mathrm{T}})\right\|^2 \qquad (14)$$

Minimizing Eq. (14) is obviously equivalent to minimizing following objective function [7].

$$\min f(\mathbf{U},\mathbf{V},\mathbf{C},\mathbf{P}_k) = \frac{1}{2}\sum_{k=1}^{K} \left\| w_k \mathbf{X}_k \mathbf{P}_k - w_k \mathbf{U} \boldsymbol{\Sigma}_k \mathbf{V}^{\mathrm{T}} \right\|^2 \quad (15)$$

Let $\mathbf{Y}_k = w_k \mathbf{X}_k \mathbf{P}_k$, $\overline{\boldsymbol{\Sigma}}_k = w_k \boldsymbol{\Sigma}_k = w_k \mathrm{diag}(\mathbf{C}_{k,\cdot})$, it is easy to see Eq. (15) corresponds to a $N \times R \times K$ cubical tensor as illustrated in the right part of Figure 4, where each slice $\mathbf{Y}_k$ has the identical size $N \times R$ (N users and R virtual items). Finally, we can obtain the full objective function for CDTF by appending the regularization terms as given by Eq. (16).

$$\min f(\mathbf{U},\mathbf{V},\mathbf{C},\mathbf{P}_k) = \frac{1}{2}\sum_{k=1}^{K} \left\| w_k \mathbf{X}_k \mathbf{P}_k - \mathbf{U}(w_k \boldsymbol{\Sigma}_k) \mathbf{V}^{\mathrm{T}} \right\|^2 + \frac{\lambda_U}{2}\|\mathbf{U}\|_F^2 + \frac{\lambda_V}{2}\|\mathbf{V}\|_F^2 + \frac{\lambda_C}{2}\|\mathbf{C}\|_F^2 \quad (16)$$

3.2.2 Algorithm

We need to reconstruct all missing values for prediction but the standard fitting algorithm for PARAFAC2 is based on the complete data [7]. Therefore, we need to design a new fitting algorithm which allows dealing with missing data. Thus, we apply an Expectation Maximization (EM) [25; 27] sub-procedure into the fitting algorithm to handle the incomplete data by iteratively imputation after each full cycle of updates.

$$\overline{\mathbf{X}}_k^{(t+1)} = \mathbf{M}_k \circledast \overline{\mathbf{X}}_k^{(t)} + (\mathbf{1} - \mathbf{M}_k) \circledast \mathbf{U}\overline{\boldsymbol{\Sigma}}_k \mathbf{V}^{\mathrm{T}} \quad (17)$$

where $\overline{\mathbf{X}}_k^{(0)} = w_k \mathbf{X}_k$ can be pre-computed and $\mathbf{M}_k$ is an indicator matrix whose entry (i,j) is one if $\mathbf{X}_k(i,j)$ has been rated (for observed values) and zero otherwise (for missing values), $\mathbf{1}$ is an all ones matrix that has the same size as $\mathbf{M}_k$.

So far we have described the detail of CDTF model and the EM algorithm for missing data handling. In summary, Algorithm 2 gives the whole factorization scheme for CDTF extending from the direct fitting algorithm for PARAFAC2 [2; 7]. In this algorithm, the computational complexity is mainly dependent on the internal sub-procedure CP. Here, we use the CP-ALS-R so the complexity is $\mathrm{O}(NRKR)$. Since the K (the number of domains) and the R (dimensionality of factor) are small, the generated tensor $\boldsymbol{\mathcal{Y}}$ can be decomposed very efficiently.

3.2.3 Recommendation

Given the estimated factor matrices $\mathbf{U}, \mathbf{V}, \mathbf{C}, \mathbf{P}_t$, the prediction of user u's rating of item i of target domain t is given by:

$$\widehat{\mathbf{X}}_t(u,i) \approx \mathbf{p}^{\mathrm{T}}\mathbf{q}$$

where $\mathbf{p}^{\mathrm{T}} = \mathbf{U}_{u,\cdot} \circledast \mathbf{C}_{t,\cdot}$ is the domain-specific user factors of u and $\mathbf{q}^{\mathrm{T}} = \mathbf{P}_t(i,\cdot)\mathbf{V}$ is the item factors of i. As a whole, the reconstructed rating matrix of target domain t is given by $\widehat{\mathbf{X}}_t \approx \mathbf{U}\boldsymbol{\Sigma}_t(\mathbf{P}_t\mathbf{V})^{\mathrm{T}}$. Let $\boldsymbol{i}$ denote the set of u's all unrated items, then we can obtain the personalized recommendation ranking over $\boldsymbol{i}$ by sorting $\widehat{\mathbf{X}}_t(u,\boldsymbol{i})$ in a descending order.

3.3 Implicit Feedback based CDTF

In real applications, the explicit feedbacks are not always available but implicit feedbacks are easily gained from user behavior history. For example, a user may not give ratings (explicit feedbacks) to the books she/he has bought but his purchase history can be considered as an implicit feedback. Consequently, some single-domain CF methods have been proposed to exploit the more abundant implicit feedbacks [6; 20; 21]. However, even implicit feedback based CF models still suffer from data sparsity and cold-start issues.

```
Algorithm 2:
[$\mathbf{U},\mathbf{V},\mathbf{C},\mathbf{P}_k$] = CDTF($\mathbf{X}_k, \mathbf{w}_k, \mathbf{M}_k, \lambda_U, \lambda_V, \lambda_C$)
Input: $\mathbf{X}_k$ the rating matrices for each domain
       $\mathbf{w}_k$ the weights for each domain
       $\mathbf{M}_k$ the indicator matrices for each slice
       $\lambda_U, \lambda_V, \lambda_C$ the regularization parameters
Output: $\mathbf{U}$ the factor matrix for users
        $\mathbf{C}$ the factor matrix for domains
        $\mathbf{V}, \mathbf{P}_k$ the factor matrices for items
Begin:
  Initialization:
1:  $\overline{\mathbf{X}}_k \leftarrow w_k\mathbf{X}_k$, $\mathbf{V} \leftarrow \mathbf{I}$, $\overline{\boldsymbol{\Sigma}}_k \leftarrow w_k\mathbf{I}$          k=1,...,K
2:  Initialize $\mathbf{U}$ principal eigenvectors $\sum_{k=1}^{K}\overline{\mathbf{X}}_k^{\mathrm{T}}\overline{\mathbf{X}}_k$ by SVD
  EM Steps:
3:  $\mathbf{Q}_k \leftarrow \overline{\mathbf{X}}_k^{\mathrm{T}}\mathbf{U}\overline{\boldsymbol{\Sigma}}_k\mathbf{V}^{\mathrm{T}}$          k=1,...,K
4:  $\mathbf{P}_k \leftarrow \mathbf{Q}_k(\mathbf{Q}_k^{\mathrm{T}}\mathbf{Q}_k)^{-\frac{1}{2}}$          k=1,...,K
5:  Generate tensor $\boldsymbol{\mathcal{Y}}$ whose slices are $\mathbf{Y}_k \leftarrow \overline{\mathbf{X}}_k\mathbf{P}_k$          k=1,...,K
6:  Update $\mathbf{U},\mathbf{V},\mathbf{C}$ by one-iteration CP-ALS-R (Algorithm 1):
        [$\mathbf{U},\mathbf{V},\mathbf{C}$] =CP-ALS-R($\boldsymbol{\mathcal{Y}}, \lambda_U, \lambda_V, \lambda_C$)
7:  $\overline{\boldsymbol{\Sigma}}_k \leftarrow \mathrm{diag}(\mathbf{C}_{k,\cdot})$          k=1,...,K
8:  $\overline{\mathbf{X}}_k \leftarrow \mathbf{M}_k \circledast \overline{\mathbf{X}}_k + (\mathbf{1} - \mathbf{M}_k) \circledast \mathbf{U}\overline{\boldsymbol{\Sigma}}_k\mathbf{V}^{\mathrm{T}}$          k=1,...,K
9:  Repeat 3 –8 until convergence
  Post Steps:
10: Rescale $\mathbf{C}_{k,\cdot} \leftarrow \frac{1}{w_k}\mathbf{C}_{k,\cdot}$ , i.e. rescale back $\overline{\boldsymbol{\Sigma}}_k$          k=1,...,K
11: Return $\mathbf{U},\mathbf{V},\mathbf{C},\mathbf{P}_k$          k=1,...,K
End
```

In particular, one-class implicit feedback is dominant in real world. For example, a one-class purchase record matrix marks entries with 1 to indicate the buy and the rest of data on this matrix are unknown. Since such one-class data is purely indiscriminate, most explicit feedback based MF/TF methods, including CDTF, cannot work well. Hence, we devised an implicit feedback enhanced CDTF (CDTF-IF) model to deal with one-class feedbacks via confidence modeling.

3.3.1 Confidence Level

In fact, implicit feedbacks can indirectly reflect opinions through user behavior because users may deliberately choose to access which items [17]. Given an observation matrix $\mathbf{R}$ of some domain, let us introduce a binary matrix $\boldsymbol{\Delta}$, where its element, $\delta_{u,p}$, indicates whether the entry $\mathbf{R}_{u,p}$ has an observed value. Note that the matrix $\mathbf{R}$ can be rating based like above, or simply all ones to indicate observed entries for one-class implicit feedbacks.

$$\delta_{u,p} = \begin{cases} 1 & \mathbf{R}_{u,p} \text{ has a value} \\ 0 & \mathbf{R}_{u,p} \text{ is missing} \end{cases}$$

$\delta_{u,p} = 1$ can be interpreted that u shows some explicit like to item p whereas $\delta_{u,p} = 0$ indicates u never consumed p, which implies u, to some extent, implicitly dislikes p. However, such implicit dislike can stem from many other reasons beyond real dislike. For example, the user might be unaware of the existence of the item, or unable to consume it due to its price [6]. Therefore, we can use varying confidence levels to represent the degree of users' like or dislike over each item.

The confidence level of user preference is proportional to the value of given rating. That is, the higher the rating is given by a user, the more the confidence indicates that the user indeed likes the item. For example, if a user rates an item with 5 (the highest score), it indicates she/he likes this item very much so we can assign a high confidence level to signify the like. On the contrary, if a user rates an item with 1 (the lowest score), it implies that she/he has much dissatisfaction with this item so a low confidence level is assigned to identify the dislike. On the other hand, to model the confidences of users' dislike over unrated items (missing values in $\mathbf{R}$), a very low confidence level is associated with these entries since we have no evidence to prove the users' explicit dislike.

According to the above analysis, we can construct a confidence matrix $\boldsymbol{\Omega}$ to indicate the level of users' like/dislike over each item:

$$\boldsymbol{\Omega}_{u,p} = \begin{cases} 1 & \delta_{u,p} = 0 \\ \alpha \mathbf{R}_{u,p} & \delta_{u,p} = 1 \end{cases} \qquad (18)$$

where $\alpha \gg 1$ is a constant to scale confidence according to the rating of items. For missing value (u, p), a small constant 1 is assigned to denote the minimal confidence of dislike.

Then, we can add the confidence matrix $\boldsymbol{\Omega}_k$ into Eq. 14 and replace the rating matrix $\mathbf{X}_k$ with indicative matrix $\boldsymbol{\Delta}_k$ for each domain. Immediately, we obtain the following objective function.

$$\min f(\mathbf{U}, \mathbf{V}_k, \boldsymbol{\Sigma}_k) = \frac{1}{2}\sum_{k=1}^{K} \left\| w_k \left[\boldsymbol{\Omega}_k \circledast \left(\boldsymbol{\Delta}_k - \mathbf{U}\boldsymbol{\Sigma}_k \mathbf{V}_k^{\mathrm{T}} \right) \right] \right\|^2 \quad (19)$$

where w_k is the weight as discussed in CDTF and $\boldsymbol{\Omega}_k$ is the confidence matrix over each domain.

3.3.2 Algorithm

Similar to the derivation of CDTF, it is possible to transform Eq. (19) into a cube based TF model by variables substitution as follows. Let $\mathbf{X}_k = (w_k \boldsymbol{\Omega}_k) \circledast \boldsymbol{\Delta}_k$ be the observation matrix and $\widehat{\mathbf{X}}_k = (w_k \boldsymbol{\Omega}_k) \circledast \left(\mathbf{U}\boldsymbol{\Sigma}_k \mathbf{V}_k^{\mathrm{T}} \right)$ be the approximate matrix, and then we substitute the variable $\widehat{\mathbf{X}}_k$ with the factorization form $\overline{\mathbf{U}}\overline{\boldsymbol{\Sigma}}_k (\overline{\mathbf{P}}_k \overline{\mathbf{V}})^{\mathrm{T}}$. So Eq. (19) can be rewritten in the form of Eq. (20) with the regularization terms appended.

$$\begin{aligned} \min f(\overline{\mathbf{U}}, \overline{\mathbf{V}}, \overline{\mathbf{C}}, \overline{\mathbf{P}}_k) &= \frac{1}{2}\sum_{k=1}^{K} \left\| \mathbf{X}_k - \overline{\mathbf{U}}\overline{\boldsymbol{\Sigma}}_k (\overline{\mathbf{P}}_k \overline{\mathbf{V}})^{\mathrm{T}} \right\|^2 \\ &+ \frac{\lambda_{\overline{U}}}{2} \|\overline{\mathbf{U}}\|_F^2 + \frac{\lambda_{\overline{V}}}{2} \|\overline{\mathbf{V}}\|_F^2 + \frac{\lambda_{\overline{C}}}{2} \|\overline{\mathbf{C}}\|_F^2 \end{aligned} \qquad (20)$$

where $\overline{\mathbf{P}}_k$ satisfies the column-wise orthonormal constraint as previously. Accordingly, the whole factorization algorithm w.r.t. factor matrices $\overline{\mathbf{U}}, \overline{\mathbf{V}}, \overline{\mathbf{C}}, \overline{\mathbf{P}}_k$ is given by Algorithm 3.

3.3.3 Recommendation

According to Eq. (19), the ranking score matrix for target domain t is computed by $\widehat{\boldsymbol{\Delta}}_t = \mathbf{U}\boldsymbol{\Sigma}_t \mathbf{V}_t^{\mathrm{T}}$. So the recommendation ranking over user u's unrated items $\boldsymbol{i}$ can be generated by descendingly sorting the $\widehat{\boldsymbol{\Delta}}_t(u, \boldsymbol{i})$. Here, the ranking score matrix $\widehat{\boldsymbol{\Delta}}_t$ can be computed by the back-substitution:

$$\widehat{\boldsymbol{\Delta}}_t = \mathbf{U}\boldsymbol{\Sigma}_t \mathbf{V}_t^{\mathrm{T}} = [\overline{\mathbf{U}}\overline{\boldsymbol{\Sigma}}_t (\overline{\mathbf{P}}_t \overline{\mathbf{V}})^{\mathrm{T}}] \oslash (w_t \boldsymbol{\Omega}_t).$$

However, such back-substitution is not necessary because w_t and $\boldsymbol{\Omega}_{t(u,p)} = 1$ are constant for all unrated items. Therefore, we can directly sort each row of $\overline{\boldsymbol{\Delta}}_t = \overline{\mathbf{U}}\overline{\boldsymbol{\Sigma}}_t (\overline{\mathbf{P}}_t \overline{\mathbf{V}})^{\mathrm{T}}$ in a descending order to rank the items for each user.

3.4 Optimal Weights Assignment

Weight parameters are also quite valuable to be discussed because they play an important role in CF models [16; 24] to control the amount of impact from auxiliary data. It has been reported in many literatures that imposing both too much and too little influence will degenerate the performance [16; 24]. So finding well-tuned weight parameters is an inevitable step to achieve better performance.

```
Algorithm 3:
[$\overline{\mathbf{U}}, \overline{\mathbf{V}}, \overline{\mathbf{C}}, \overline{\mathbf{P}}_k$] = CDTF-IF($\boldsymbol{\Delta}_k, \mathbf{w}_k, \boldsymbol{\Omega}_k, \lambda_U, \lambda_V, \lambda_C$)
Input: $\boldsymbol{\Delta}_k$ is the indicative matrices for each domain
       $\mathbf{w}_k$ is the weight for each domain
       $\boldsymbol{\Omega}_k$ is the confidence matrix for each slice
       $\lambda_{\overline{U}}, \lambda_{\overline{V}}, \lambda_{\bar{C}}$ are the regularization parameters
Output: $\overline{\mathbf{U}}$ is the factor matrix for users
        $\bar{\mathbf{C}}$ is the factor matrix for domains
        $\overline{\mathbf{V}}, \overline{\mathbf{P}}_k$ is the factor matrices for items
Begin:
  Initialization:
  1: $\mathbf{X}_k = (w_k \boldsymbol{\Omega}_k) \circledast \boldsymbol{\Delta}_k$, $\mathbf{V} \leftarrow \mathbf{I}$, $\overline{\boldsymbol{\Sigma}}_k \leftarrow \mathbf{I}$        k=1,...,K
  2: Initialize $\overline{\mathbf{U}}$ principal eigenvectors $\sum_{k=1}^{K} \mathbf{X}_k^{\mathrm{T}} \mathbf{X}_k$ by SVD
  Iteration:
  3: $\mathbf{Q}_k \leftarrow \mathbf{X}_k^{\mathrm{T}} \overline{\mathbf{U}}\overline{\boldsymbol{\Sigma}}_k \overline{\mathbf{V}}^{\mathrm{T}}$        k=1,...,K
  4: $\mathbf{P}_k \leftarrow \mathbf{Q}_k \left( \mathbf{Q}_k^{\mathrm{T}} \mathbf{Q}_k \right)^{-\frac{1}{2}}$        k=1,...,K
  5: Construct tensor $\boldsymbol{\mathcal{Y}}$ whose slices are $\mathbf{Y}_k \leftarrow \mathbf{X}_k \mathbf{P}_k$        k=1,...,K
  6: Update $\overline{\mathbf{U}}, \overline{\mathbf{V}}, \bar{\mathbf{C}}$ by one-iteration CP-ALS-R (Algorithm 1):
         [$\overline{\mathbf{U}}, \overline{\mathbf{V}}, \bar{\mathbf{C}}$] =CP-ALS-R($\boldsymbol{\mathcal{Y}}, \lambda_{\overline{U}}, \lambda_{\overline{V}}, \lambda_{\bar{C}}$)
  7: $\overline{\boldsymbol{\Sigma}}_k \leftarrow \mathrm{diag}\left( \bar{\mathbf{C}}_{k,\cdot} \right)$        k=1,...,K
  Repeat 3 –7 until convergence
  8: Return $\overline{\mathbf{U}}, \overline{\mathbf{V}}, \bar{\mathbf{C}}, \overline{\mathbf{P}}_k$        k=1,...,K
End
```

The same holds true for CDTF and CDTF-IF, where the weights assigned on each domain exactly act as such trade-off parameters. Too large weights assigned to auxiliary domains may overwhelm the information from target domain while too small weights may fail to transfer enough knowledge to target domain.

Most of current CF methods [16; 24] usually only involve one or two auxiliary relations so a common way to find an optimal model is to select the best setting from a group of manually given values via cross validation. However, the CDCF problem often involve several domains and the number of possible weights assignments grows with the number of domains exponentially. For example, if we have four domains and the weight on each domain has five possible values $\{0.01, 0.1, 1, 10, 100\}$, then the possible number of combinations of weights is 5^4. Obviously, it will be a painful process to find the optimal weights assignment in semi-manual way by cross validation. Moreover, such heuristically pre-given values do not guarantee to cover real optimal values. Hence, we need a more robust and automated method to find the best weights assignment.

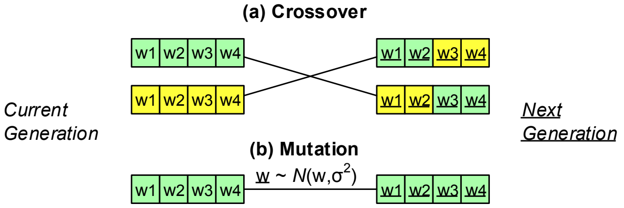


Figure 5: Searching optimal weights assignment by crossover and mutation operators using GA

Here, we employ the genetic algorithm (GA) [3] to find such optimal weights assignment. We fix the weight on target domain to be 1 so various weights assignments on auxiliary domains act as the individuals in the population. In GA the crossover operator combines a part of elements in each parent to form children for the next generation, so it enables the automatic search for the best combination of weights as depicted in Figure 5(a). And the mutation operator applies random changes to a single individual in the current generation to create a child. As illustrated in Figure 5(b), the weights in each generation are automatically adjusted by mutation.

Since the range of weight is large, a uniformly randomized initial population will take too long time to converge. So we take the following strategy to initialize the individuals with exponential growth, where $\alpha \in (0,1]$ is a constant to scale weight, β and γ are integers to control the range of weight, $\mathbf{1}$ is an all-one vector with the length equal to the number of auxiliary domains.

$$\boldsymbol{w}_i^{\alpha} = \left(\alpha \times 10^{i}\right) \times \mathbf{1} \qquad \text{for } i = \beta, \cdots, \gamma$$

Taking an example, we initialize the population as $\boldsymbol{w} = \{\boldsymbol{w}^{0.5}, \boldsymbol{w}^{1}\}$, with $\beta = -2$ and $\gamma = 2$, and then we obtain 10 weight vectors (i.e. individuals) ranging from 0.005 to 100.

Accordingly, the change caused by mutation should match with the order of weight instead of fluctuating in the entire range, so we set up the following mutation rule to generate the child weight $w_{(g+1)}$, where G specifies the maximum number of generations and $\mathcal{N}(\cdot)$ denotes a normal distribution with the mean $w_{(g)}$ and the standard deviation σ shrinking as generations go by.

$$w_{(g+1)} \sim \mathcal{N}\left(w_{(g)} \middle| \left(w_{(g)}\sigma_{(g+1)}\right)^{2}\right), \;\; s.t. \, w_{(g+1)} > 0$$

$$\sigma_{(g+1)} = 1 - \frac{(g+1)}{G}$$

Following the above initial strategy and mutation rule, we can run the GA given any fitness function, e.g. the MAE (mean absolute error) over the testing dataset using CDTF.

4. EXPERIMENTS

The experiments are conducted on two real world datasets, namely the ratings of Amazon products and the follow records of a social networking site. In the following experiments, we evaluated the performance of the rating and ranking prediction by a set of metrics and compared our models with other state-of-the-art approaches.

4.1 Amazon Data

Amazon[1] is the most famous e-business website to sell diverse products, such as books, DVDs, shoes, etc. The dataset [10] was crawled from Amazon website and it contains 1,555,170 users and 1-5 scaled ratings over 548,552 different products covering four domains: 393,558 books, 103,144 music CDs, 19,828 DVDs and 26,132 VHS video tapes. Obviously, the users' preferences are dependent across these domains, so it is very suitable to test CDCF algorithms over this dataset.

Data Preparation: In this experiment, we selected *Book* and *Music CD* as the target domain to evaluate respectively. We filtered out users who have rated at least 50 books or 30 music CDs so that there are enough observations to be split in various proportions of training and testing data for our evaluation. Finally, 2,505 users were selected, and in addition we retrieved all items rated by these users in these four domains and set aside top *K* rated items for each domain respectively. Table 1 shows the statistics of the data for evaluation. Then, we constructed rating matrices over filtered out data for each domain.

- *Sparse Data Case:* To simulate the sparse data problem, we constructed two sparse training sets, ***TR₂₀*** and ***TR₇₅***, by respectively holding out 80% and 25% data from the target domain *Book*, i.e. the remaining data of target domain for training is 20% and 75%. The hold-out data servers as ground truth for testing. Likewise, we also construct two other training sets ***TR₂₀*** and ***TR₇₅*** when choosing *Music* as the target domain.
- *Unacquainted World Case*: We randomly select half users and hold out all their data from target domain to simulate the unacquainted world phenominon. The training set used for this case is denoted as ***TRᵤᵥᵥ***.

Table 1. Statistics of amazon data for evaluation

Domain	*Items*	*Avg. # ratings for each item*	*Avg. # ratings for each user*	*Density*
Book	6000	24	57	0.0097
Music	5000	15	30	0.0062
DVD	3000	30	37	0.0124
VHS	3000	29	35	0.0117

Methods: In all following experiments, a group of state-of-the-art methods are evaluated for comparison, including our models. When running the evaluation using each compared method, we set the dimensionality of factors and other hyper-parameters by cross validation.

- *MF-SGD:* It is a single domain based MF model to minimize the squared error by stochastic gradient descent [9]. It directly takes the rating matrix of the target domain as input and cannot handle the cold-start problem.
- *N-CDCF-U:* A user-based neighborhood CDCF model uses Eq. (1) for prediction. In this experiment, we use k=10 closest users.
- *N-CDCF-I:* An item-based neighborhood CDCF model uses Eq. (3) for prediction. In this experiment, we use k=10 closest items.
- *MF-CDCF:* A MF model, described in Section 2, takes the long concatenated rating matrix as input.
- *CMF:* Collective matrix factorization [24] is a CDMF which couples rating matrices for all domains on the *User* dimension so as to transfer knowledge through the common user-factor matrix
- *CDTF:* Our model, which is described in Section 3.2, takes one of the above domains as target domain to perform prediction and others as auxiliary domains to borrow knowledge.
- *PF2-CDCF:* The main difference of PARAFAC2 [4] from our CDTF is that it does not have the mechanism to adjust the amount of knowledge contributed by each domain.

Metrics: we used the most widely used evaluation metric for CF problem, namely *Mean Absolute Error* (MAE) [26], to measure the rating prediction quality.

$$\text{MAE} = \sum\nolimits_{r_{u,p} \in \boldsymbol{TS}} \frac{abs\left(r_{u,p} - \hat{r}_{u,p}\right)}{N}$$

where $r_{u,p}$ denotes the true rating user u gave to item p, $\hat{r}_{u,p}$ is the predicted rating, and N denotes the number of ratings in testing set.

Comparison: We evaluated the prediction performance using three differently sparse training sets, namely ***TR₇₅***, ***TR₂₀*** and ***TRᵤᵥᵥ*** constructed above. Table 2 reports the evaluation results with setting *Book* and *Music CD* as target domains respectively.

[1] http://www.amazon.com/

Table 2. MAE (the smaller the better) of comparative models with different training sets

Models	*Target Domain: Book*			*Target Domain: Music*		
	TR_{75}	TR_{20}	TR_{uw}	TR_{75}	TR_{20}	TR_{uw}
MF-SGD	0.597	0.833	---	0.749	0.942	----
N-CDCF-U	0.488	0.776	0.843	0.701	0.906	0.907
N-CDCF-I	0.728	0.850	0.826	0.776	1.062	0.873
MF-CDCF	0.503	0.753	0.855	0.715	0.832	0.879
CMF	0.452	0.764	0.821	0.686	0817	0.848
PF2-CDCF	0.507	0.765	0.800	0.709	0.827	0.839
CDTF	**0.327**	**0.672**	**0.757**	**0.664**	**0.751**	**0.776**

From Table 2, we can find that most CDCF models achieve much better performance than single-domain CF model. Therein, our model, CDTF, significantly outperforms all other comparative CDCF methods over all testing sets. Especially, more than 35% improvement is achieved in the case of $\boldsymbol{TR_{75}}$ training sets, which illustrates that CDTF can better capture personalized factors for each domain when users' feedbacks are relatively sufficient. N-CDCF-U also achieves a not bad performance when the data is relative dense, i.e. $\boldsymbol{TR_{75}}$, but the performance decreases very fast when the data becomes sparser. As analyzed in Section 2, such neighborhood based method usually fails to find global similarity among users when the data is sparse. The typical CDMF model, CMF, overall outperforms MF-CDCF. It is because that CMF provides a more effective way to transfer knowledge between domains instead of ignoring the heterogeneity between domains and integrating all data into a single matrix as MF-CDCF.

Not surprisingly, CDTF achieves the best performance again in the unacquainted-world cases, i.e. using the training sets $\boldsymbol{TR_{uw}}$, which can be mainly attributed to the triadic relation modeling over *user-item-domain* so CDTF can better recover the domain-specific user preferences than other models. In comparison, CMF lags much behind CDTF. The reason is that CMF only models a couple of dyadic relations over users and items like traditional MF models so, for the unacquainted-world cases, it cannot learn the domain-specific user factors due to the absence of domain factors. Specially, PF2-CDCF cannot achieve a good performance though it is also a TF model based on triadic relation. The main reason is that it does not provide a mechanism to trade off the amount of influence contributed from each domain. Next, we will demonstrate impact of weights assignment on the prediction results.

Impact of Weights: CDTF offers a flexible mechanism to balance the amount of influence among auxiliary domains by tuning the weight assigned to each domain. For example, we selected *Book* as the target domain and varied the weight on all other auxiliary domains from 0.01 to 100 exponentially and report the MAEs over $\boldsymbol{TR_{20}}$ in Figure 6 (a). We can find that the performance is quite different with different weights assignments. To find the optimal weights assignment, we ran the GA with initial population $\boldsymbol{w} = \{\boldsymbol{w}^{0.33}, \boldsymbol{w}^{0.66}, \boldsymbol{w}^{1}\}$ and $\beta = -2$, $\gamma = 2$, i.e. there are totally 15 initial individuals with different scale. The rightmost bar in Figure 6 (a) shows the optimal result through GA. Obviously, it performs much better than those results with heuristically setting weights. Figure 6 (b) depicts the MAEs changing with iterations, and we can find that it converges very fast and reaches the optimal result within 10 generations.

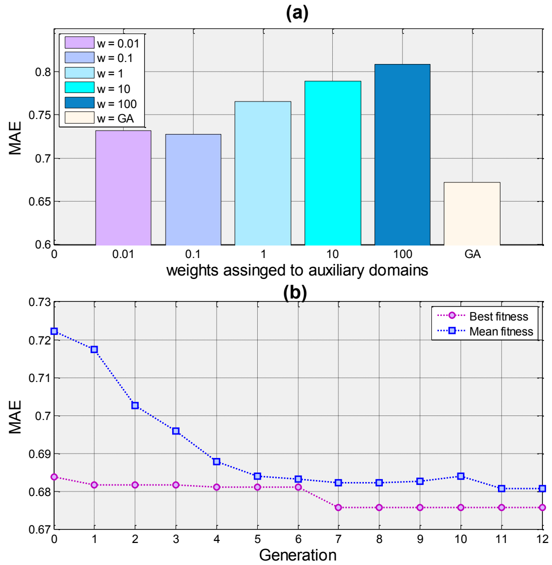


Figure 6: (a) Comparison of weights assignment optimized by GA to the weights settings varying from 0.01 to 100; (b) The converging status over generations for searching the optimal weights assignment.

Table 3 reports the results of optimal weights assignments learned through GA. These results prove an obvious truth that the target domain needs only a little information transferred from auxiliary domains (relative small weights on auxiliary domains) if there are sufficient data on it, but it needs to leverage more and more information from auxiliary domains (relative large weights on auxiliary domains) when the data become sparser.

Table 3. Optimal weights assignments found through GA over six different training sets

Weight	*Target Domain: Book*			*Target Domain: Music*		
	TR_{75}	TR_{20}	TR_{uw}	TR_{75}	TR_{20}	TR_{uw}
w_{Book}	1	1	1	0.012	0.336	1.444
w_{Music}	0.001	0.012	13.64	1	1	1
w_{DVD}	0.002	0.019	3.940	0.001	0.891	0.923
w_{VHS}	0.009	0.135	1.875	0.020	0.459	0.030

4.2 CDCF in Social Networking Site

Social networking sites (SNS) may be the most successful product in the age of Web 2.0. People share videos, blogs, games, movies reviews and all kinds of things on SNS but most people only focus their interests on very a few domains. Finding possible attractive items in people's unacquainted domains cannot only improve user experience but bring more profits. In a SNS, we often only know what items users explicitly like, e.g. the applications added or the groups followed, but no information about what they explicitly dislike. Therefore, it is suggested to perform the implicit feedback CDCF algorithm.

In this experiment, we use the SNS dataset provided by KDD Cup[2], where the dataset provides the profile for each user about the items she/he has followed. These items are categorized into different domains with anonymous names which may be interpreted as game, sports, entertainment, etc. Hence, we ran CDCF methods over this dataset to evaluate the performance of item ranking.

[2] http://www.kddcup2012.org/c/kddcup2012-track1

Data Preparation: Four domains accounted for most data from the dataset, namely A, B, C and D, are selected for evaluation. We filtered out 7,000 users who have followed at least 10 items in domain A and B. Then, we obtain a dataset that contains about one million follow records over above four selected domains. These follow records are representing one-class valued matrix for each domain, that is, the entry with value 1 denotes a user has followed this item. Table 4 illustrates the statistics of this dataset.

Table 4. Statistics of SNS data for evaluation

Domain	*Items*	*Avg. # users following an item*	*Avg. # items a user following*	*Density*
A	859	144	17	0.0206
B	313	287	12	0.0407
C	863	487	60	0.0681
D	329	251	11	0.0360

In this experiment, domain A and B are chosen as the target domains for evaluation respectively. For each target domain, we apply the same strategy to construct two training sets $\boldsymbol{TR_{25}}$ and $\boldsymbol{TR_{uw}}$ as done in the first experiment to evaluate the sparse-data and unacquainted-world case respectively. All the hold-out data are used as ground truths for testing.

Methods: Some models in the first experiment do not support implicit feedbacks, so they cannot handle this dataset consisting of one-class values. We provide some other methods instead.

- *Most-Pop:* It is a most widely used strategy to rank item by its popularity (measured by the number of users following).
- *N-CDCF-U:* Given a user, the similarity with other users is computed by cosine similarity over all items (using binary ratings). Then, the ranking score is computed as a weighted average rating over all other users. In this experiment, we use k=10 closest neighbors.
- *MF-CDCF-IF:* The concatenated matrix based CDCF model described in Section 2, taking advantage of implicit feedback [6].
- *CDTF-IF:* Our cross-domain factorization model using implicit feedback, described in Section 3.3.
- *PF2-IF:* We extend the original PARAFAC2 [4] to support implicit feedbacks as applying to CDTF-IF. The main difference from CDTF-IF is that PF2-IF does not have weight parameters to trade off the influence from each domain.

Metrics: We use two frequently used metrics, *recall* and *AUC* [21; 26], to evaluate the quality of ranking for recommendation over $\boldsymbol{TS_u}$, i.e. the positively followed items for each user in testing set.

- *recall@N* considers the positively followed items within the top *N*, a high recall with lower *N* will be a better system:

$$recall@N = \frac{\#hits@N}{|\boldsymbol{TS_u}|}$$

- *AUC* **(**Area under the ROC curve) measures the probability that a system ranks a positive instance higher than a negative one.

$$AUC = \frac{\sum_{i \in \boldsymbol{TS_u}} \sum_{k \in \boldsymbol{I} \setminus \boldsymbol{TS_u}} \delta\big(rk(i) < rk(k)\big)}{|\boldsymbol{TS_u}| \cdot |\boldsymbol{I} \setminus \boldsymbol{TS_u}|}$$

where $\boldsymbol{I}$ denotes all items in the target domain, $rk(i)$ retrieve the rank of item i created by some model and $\delta\big(rk(i) < rk(k)\big)$ is the delta function returning 1 if $rk(i) < rk(k)$ and 0 otherwise.

Below we report the results using the average *recall* and *AUC* from all testing users.

Comparison: The performance of prediction is evaluated using two training sets $\boldsymbol{TR_{25}}$, $\boldsymbol{TR_{uw}}$ with setting domain A and B as the target domain respectively. For our model CDTF-IF, the scaling constant in Eq. 18 is set as $\alpha = 10$ and the weight parameters are automatically optimized by GA. Table 5 reports the results of all comparative methods using the metric AUC.

Table 5. AUC (the larger the better) of comparative models over different training sets

Model	***Target Domain: A***		***Target Domain: B***	
	$\boldsymbol{TR_{25}}$	$\boldsymbol{TR_{uw}}$	$\boldsymbol{TR_{25}}$	$\boldsymbol{TR_{uw}}$
Most-Pop	**0.8391**	0.9317	0.8015	0.9389
N-CDCF-U	0.5015	0.8112	0.6210	0.8122
MF-CDCF-IF	0.8388	0.9103	0.7980	0.9276
PF2-IF	0.8205	0.6832	0.7276	0.7358
CDTF-IF	0.8365	**0.9570**	**0.8069**	**0.9533**

A little surprisingly, the Most-Pop method performs better than all other models except CDTF-IF. Through a further consideration, we concluded it reveals a general fact that the hot events, music, movies, tweets, etc. are usually listed on the home pages of SNS so users will actively or passively keep their eyes on these popular items and share them with their friends. Such "rich get richer" phenomenon over items is ubiquitous on SNS so it leads to a high AUC. The neighborhood based method, N-CDCF-U, does not perform very well due to the inherent weakness of finding the closest users over the sparse data. CDTF-IF surpassing Most-Pop proves that CDTF-IF not only concerns popular items but also better captures personalized preferences.

The metric recall@k can effectively check if a recommender system can successfully retrieve the items that user has shown positive preference by comparing the top k recommended items from its returned list. Hence we evaluate recall@5~100 over all comparative methods. Figure 7 reports the results over four different training sets. Most-Pop does not achieve a high recall@k when k is relative small, which illustrates that apart from some most popular items people tend to follow much more personalized favorite items. Similar to Most-Pop, N-CDCF-U also depends on other users' preferences so its performance is close to Most-Pop. In particular, PF2-IF lags behind CDTF-IF due to the lack of adjusting the appropriate amount of influence among target domain and auxiliary domains. Obviously, as illustrated in all four figures, the plots of CDTF-IF are consistently above those of all other models, so we can conclude that CDTF-IF can better capture the domain-specific personalized preference than other models.

5. RELATED WORK

Most state-of-the-art CDCF models are extended from single-domain MF models, where knowledge from auxiliary domains are transferred into target matrix by some shared factor matrices.

Codebook Transfer [12] assumes some cluster-level rating patterns, which are represented by a codebook, can be found between the rating matrices in two related domains. Rating-Matrix Generative Model [13] extends this idea with a probabilistic model to solve collective transfer learning problems. In reality, there are many cold-start users for most domains. Therefore, it is always out of the question to find common patterns when the user data is absent in some domain, i.e. the unacquainted world case. Dual Transfer Learning (DTL) [14] exploits the duality between by matrix tri-factorization, which mainly aims to solve the clustering or classification problems. Given observed features of source and target domains, the marginal distribution corresponds to common latent features learned from the data over all domains, and the conditional distributions corresponds to domain-specific latent features. However, the explicit features are commonly not available in collaborative filtering so DTL is not applicable to the CDCF problems as studied in this paper.

Since the user preference is not exclusive in a single domain, a more straightforward way is to transfer knowledge through the user-factor matrix. Collective matrix factorization (CMF) [24] couples target matrix and all auxiliary matrices on *User* dimension to share the user factor matrix across all domains. Similar to our model, CMF assigns a weight to the loss of fitting each matrix so that it can control the amount of influence from each domain on the user factor matrix. However, CMF does not provide a mechanism to find an optimal weights assignment. Coordinate System Transfer (CST) [19] learns the user-factor matrix $\mathbf{U}_\mathrm{A}$ from an auxiliary rating matrix in the first step, and then generates the user-factor matrix $\mathbf{U}_\mathrm{T}$ for the target domain based on $\mathbf{U}_\mathrm{A}$, with the regularization of penalizing the divergence between $\mathbf{U}_\mathrm{A}$ and $\mathbf{U}_\mathrm{T}$. As pointed at the beginning, CST cannot be applied to the multiple domains (more than two) scenarios as studied in this paper. Furthermore, all above models inevitably suffer from the unacquainted-world issue, which may lead to very poor recommendations.

Hu et al. [6] presented an implicit feedback based MF model, where they used a similar strategy with us to assign a small confidence to unrated items to represent implicit dislike. Bayesian personalized ranking (BPR) [20] treats explicit rated item as positive class and unrated item as negative class and it uses those classified items to represent the preference ordering between each pair of items. However, it degenerates into a one-class feedbacks problem (all negative items) for cold-start users. Hence, no single-domain model can work. It will lead to a typical unacquainted-world problem if we resolve such cold-start implicit feedback problem using CDMF models. To the best our knowledge, we are the first to introduce implicit feedbacks into the CDCF problem.

In summary, current CDMF methods generally cannot address the unacquainted-world issue because they only model the user and item factors but absolutely ignore the domain factors, whereas our CDTF models introduce the triadic interaction among user, item and domain factors so as to overcome the unacquainted-world issue.

6. CONCLUSION

In this paper, we have discussed the requirements of CDCF in current web era and limitations of current CDCF methods in an unacquainted world. Our triadic relational CDCF solution, namely CDTF and CDTF-IF, is proposed. The experiments have evaluated the performance of rating and ranking prediction in terms of various metrics using our models and other comparative methods. The evidence from all results has shown that our cross-domain factorization models significantly outperform all other state-of-the-art methods, especially for cold-start cases. It is because our tensor based models can better capture the triadic relation between users, items and domains than only the dyadic relation between users and items modeled by other methods, which lose the domain-specific information. The experiments also proved the efficiency of our GA algorithm to find an optimal weights assignment, which can achieve a much better prediction than PARAFAC2.

In the future, we may extend our model to be a time-varying CF model, since the number of items is always changing over time and users can give feedbacks to the same item multiple times. Therefore, we can create a feedback matrix for each time stamp so as to construct a TF model like CDTF. Such model can better capture the temporal factors and the shift of item factors over time.

7. ACKNOWLEGEMENTS

This work is partially supported by China National Science Foundation (Granted Number 61073021 and 61272438), Research Funds of Science and Technology Commission of Shanghai Municipality (Granted Number 11511500102 and 12511502704) and Cross Research Fund of Biomedical Engineering of Shanghai Jiaotong University (YG2011MS38).

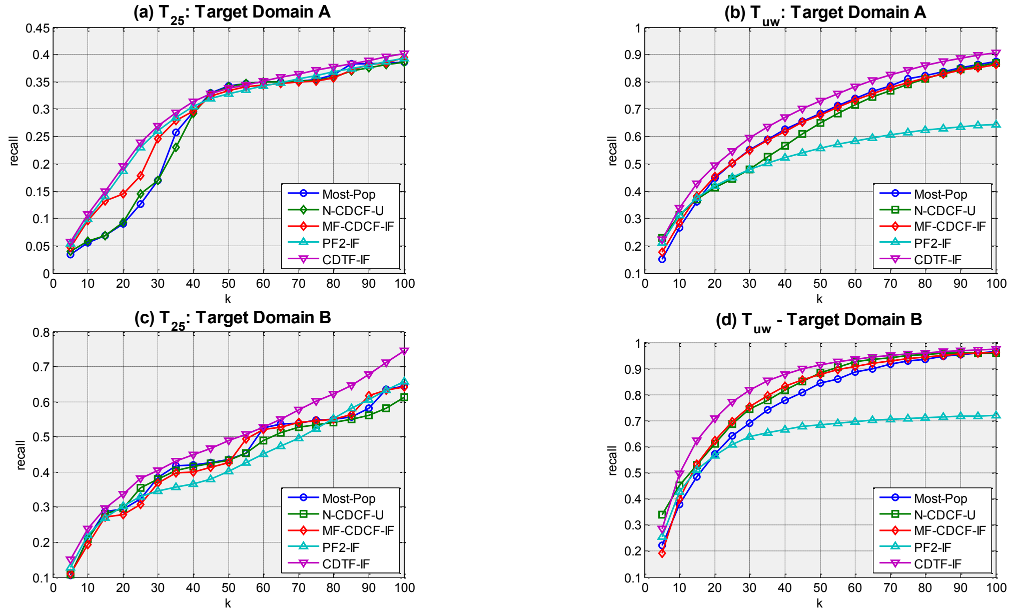


Figure 7: Comparison of CDTF-IF to Most-Pop, N-CDCF-U, MF-CDCF-IF and PF2-IF using the metrics *recall@5~100* : (a) T_{25} w.r.t Target Domain A; (b) T_{uw} w.r.t Target Domain A; (c) T_{25} w.r.t Target Domain B; (d) T_{uw} w.r.t Target Domain B.

8. REFERENCES

[1] Berkovsky, S., Kuflik, T., and Ricci, F., 2007. Cross-Domain Mediation in Collaborative Filtering. In *Proceedings of the 11th international conference on User Modeling* Springer-Verlag, Corfu, Greece, 355-359.

[2] Bro, R., 1998. Multi-way analysis in the food industry: models, algorithms, and applications University of Amsterdam

[3] Goldberg, D., 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional.

[4] Harshman, R.A., 1972. PARAFAC2: Mathematical and technical notes. *UCLA Working Papers in Phonetics 22*, 30--44.

[5] Hofmann, T., 2004. Latent semantic models for collaborative filtering. *ACM Trans. Inf. Syst. 22*, 1, 89-115.

[6] Hu, Y., Koren, Y., and Volinsky, C., 2008. Collaborative filtering for implicit feedback datasets IEEE, 263-272.

[7] Kiers, H.A.L., ten Berge, J.M.F., and Bro, R., 1999. PARAFAC2—Part I. A direct fitting algorithm for the PARAFAC2 model. *Journal of Chemometrics 13*, 3-4, 275-294.

[8] Kolda, T.G. and Bader, B.W., 2009. Tensor decompositions and applications. *SIAM review 51*, 3, 455-500.

[9] Koren, Y., Bell, R., and Volinsky, C., 2009. Matrix factorization techniques for recommender systems. *Computer 42*, 8, 30-37.

[10] Leskovec, J., Adamic, L.A., and Huberman, B.A., 2007. The dynamics of viral marketing. *ACM Trans. Web 1*, 1, 5.

[11] Li, B., 2011. Cross-Domain Collaborative Filtering: A Brief Survey. In *Proceedings of the 2011 IEEE 23rd International Conference on Tools with Artificial Intelligence* IEEE Computer Society, 1085-1086.

[12] Li, B., Yang, Q., and Xue, X., 2009. Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. In *IJCAI* Morgan Kaufmann Publishers Inc., 2052-2057.

[13] Li, B., Yang, Q., and Xue, X., 2009. Transfer learning for collaborative filtering via a rating-matrix generative model. In *Proceedings of the 26th Annual International Conference on Machine Learning* ACM, Montreal, Quebec, Canada, 617-624.

[14] Long, M., Wang, J., Ding, G., Cheng, W., Zhang, X., and Wang, W., 2012. Dual transfer learning. In *Proceedings of the 12th SIAM International Conference on Data Mining*

[15] Mørup, M., 2011. Applications of tensor (multiway array) factorizations and decompositions in data mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 1*, 1, 24-40.

[16] Ma, H., Yang, H., Lyu, M.R., and King, I., 2008. SoRec: social recommendation using probabilistic matrix factorization. In *Proceeding of the 17th ACM conference on Information and knowledge management* ACM, Napa Valley, California, USA, 931-940.

[17] Marlin, B.M., Zemel, R.S., Roweis, S., and Slaney, M., 2007. Collaborative filtering and the missing at random assumption. In *Proceeding 23rd Conference on Uncertainty in Artificial Intelligence*

[18] Pan, W., Xiang, E.W., Liu, N.N., and Yang, Q., 2010. Transfer learning in collaborative filtering for sparsity reduction. In *AAAI*

[19] Pan, W., Xiang, E.W., Liu, N.N., and Yang, Q., 2010. Transfer learning in collaborative filtering for sparsity reduction. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*

[20] Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L., 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence* AUAI Press, Montreal, Quebec, Canada, 452-461.

[21] Rendle, S., Freudenthaler, C., and Schmidt-Thieme, L., 2010. Factorizing personalized Markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web* ACM, Raleigh, North Carolina, USA, 811-820.

[22] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J., 1994. GroupLens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work* ACM, Chapel Hill, North Carolina, United States, 175-186.

[23] Schein, A.I., Popescul, A., Ungar, L.H., and Pennock, D.M., 2002. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* ACM, Tampere, Finland, 253-260.

[24] Singh, A.P. and Gordon, G.J., 2008. Relational learning via collective matrix factorization. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* ACM, Las Vegas, Nevada, USA, 650-658.

[25] Srebro, N. and Jaakkola, T., 2003. Weighted low-rank approximations. In *Proceedings of the Twentieth International Conference on Machine Learning*, Washington DC, 720.

[26] Su, X. and Khoshgoftaar, T.M., 2009. A survey of collaborative filtering techniques. *Adv. in Artif. Intell. 2009*, 2-2.

[27] Tomasi, G. and Bro, R., 2005. PARAFAC and missing values. *Chemometrics and Intelligent Laboratory Systems 75*, 2, 163-180.