Broad Learning based Multi-Source Collaborative Recommendation

Junxing Zhu[1], Jiawei Zhang[2], Lifang He[3], Quanyuan Wu[1],
Bin Zhou[1], Chenwei Zhang[4], Philip S. Yu[4,5]
[1]College of Computer, National University of Defense Technology, Changsha, Hunan, China
[2]IFM Lab, Department of Computer Science, Florida State University, Tallahassee, FL, USA
[3]Institute for Computer Vision, Shenzhen University, Shenzhen, Guangdong, China
[4]University of Illinois at Chicago, Chicago, IL, USA
[5]Institute for Data Science, Tsinghua University, Beijing, China
daishanshen5036@sina.com,jzhang@cs.fsu.edu,lifanghescut@gmail.com,qywu@nudt.edu.cn,binzhou@nudt.edu.cn,
czhang99@uic.edu,psyu@cs.uic.edu

ABSTRACT

Anchor links connect information entities, such as entities of movies or products, across networks from different sources, and thus information in these networks can be transferred directly via anchor links. Therefore, anchor links have great value to many cross-network applications, such as cross-network social link prediction and cross-network recommendation. In this paper, we focus on studying the recommendation problem that can provide ratings of items or services. To address the problem, we propose a *Cross-network Collaborative Matrix Factorization (CCMF)* recommendation framework based on broad learning setting, which can effectively integrate multi-source information and alleviate the sparse information problem in each individual network. Based on item anchor links, *CCMF* can fuse item similarity information and item latent information across networks from different sources. And different from most of the traditional works, *CCMF* can make multi-source recommendation tasks collaborate together via the information transfer based on the broad learning setting. During the transfer process, a novel cross-network similarity transfer method is applied to keep the consistency of item similarities between two different networks, and a domain adaptation matrix is used to overcome the domain difference problem. We conduct experiments to compare the proposed *CCMF* method with both classic and state-of-the-art recommendation techniques. The experimental results illustrate that *CCMF* outperforms other methods in different experimental circumstances, and has great advantages on dealing with different data sparse problems.

CCS CONCEPTS

•**Information systems → Data mining;** *Collaborative and social computing systems and tools; Information integration;*

KEYWORDS

Anchor links; recommendation; cross-network; matrix factorization

CIKM'17, November 6–10, 2017, Singapore.
© 2017 ACM. ISBN 978-1-4503-4918-5/17/11…$15.00
DOI: https://doi.org/10.1145/3132847.3132976

1 INTRODUCTION

In the real-world, information entities are extensively connected to each other via various kinds of links [28]. For instance, the co-author links connect the researchers who have collaborated before in bibliographic networks [19]; the supervisor links connect managers to their subordinates in enterprise organizational charts [28]; and the anchor links connect information entities of common items/users from different network sources (e.g., if an entity in network A and an entity in network B both represent the same item, the link which connects these two entities is an anchor link) [30]. However, different from most of the other links, anchor links normally follow the *one-to-one constraint* [5], i.e., each item/user can have at most one information entity to represent it in each network. We note the case that items/users have multiple information entities in one network is a different problem [1], and can be resolved with methods like [20], where these duplicated entities can be aggregated in advance to form one unique entity and the constraint on anchor links connecting these formed entities will still be "one-to-one".

Since anchor links are the inter-network links that connect information entities of the same users or items in two different network sources and follow the *one-to-one constraint*, the information of these connected entities in different sources/domains can be directly transferred via the anchor links. Thus anchor links can help overcome the *negative transfer* problem [21]. This problem is caused by transferring information between the domains that are not related enough, and is still not well solved by most of the current transfer learning works, where these works focus on "What to transfer" and "How to transfer", by implicitly assuming that the source and target domains are related to each other [21]. On this basis, anchor links can improve the effect of cross-network information transfer, and have great value to many cross-network applications. As a result, how to apply anchor links to cross-network applications becomes a new problem, and is explored by several works recently, which include: cross-network user alignment [5–8, 27, 35], cross-network social link prediction [29, 30], cross-network recommendation [13, 23, 24].

We are living in an era with explosive information, and digital revolution has changed the way we access information. People

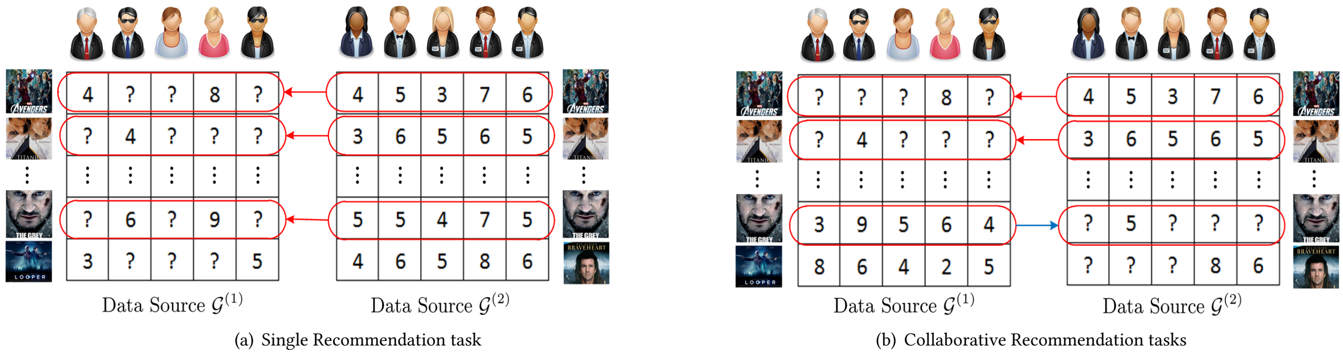


Figure 1: Two application circumstances of using anchor links to improve the recommendation performances in networks from different sources.

nowadays can easily be overwhelmed by a vast amount of candidate information entities on the Internet [22]. In order to recommend people with the information entities that match their interests, many recommendation methods have been proposed so far. However, the performances of traditional recommendation methods are usually restricted by the problem of information sparsity. For example, in IMDb[1], most of the users only post a very small number of review comments for the movies they have watched. Based on such limited information, it is very challenging for the service provider to provide high quality recommendation services for these users.

Fortunately, besides the user feedback information (e.g., the user-item rating information), there also exists many other auxiliary information that can be used to help solve the information sparsity problem [25]. And many recommendation methods based on auxiliary information have been proposed so far [3, 9, 13, 15, 21, 22]. However, most of them [9, 13, 15, 22] aims at utilizing the auxiliary information contained in the same data source where the recommendation task is conducted (e.g., utilizing the user relations in a given data source to help recommending the items in the same data source [9]). And when this kind of auxiliary information is insufficient or unavailable in the data source, the information sparsity is still a big problem. E.g., in the real world e-commerce sites (like Amazon) or recommendation sites (like Yelp), few social relations really exist among users, so the social relation information cannot be used to solve the information sparsity. One novel way to overcome such problem is to connect two data sources together with the broad learning setting. In the broad learning setting, for each data source if the useful information (includes the user feedback information and other auxiliary information) is insufficient or unavailable, some information from the other network can be transferred to help the recommendation task. Fig. 1 illustrates two application circumstances of using anchor links to overcome the information sparsity problem.

Here, "**Broad Learning**" is a new type of learning task, which focuses on fusing multiple large-scale information sources of diverse varieties together and carrying out synergistic data mining tasks across these fused sources in one unified analytic [31–34]. In the real world, on the same information entities, e.g., social media users [31, 34], movie knowledge library entries (studied in this paper) and employees in companies [32, 33], a large amount of information can actually be collected from various sources. These sources are usually of different varieties, like Foursquare vs Twitter [31, 34], IMDB vs Douban Movie sites (studied in this paper), Yammer vs company organizational chart [32, 33]. Each information source provides a specific signature of the same entity from a unique underlying aspect. Effective fusion of these different information sources provides an opportunity for researchers and practitioners to understand the entities more comprehensively, which renders "**Broad Learning**" an extremely important learning task. Fusing and mining multiple information sources of large volumes and diverse varieties are a fundamental problem in big data studies. "**Broad Learning**" investigates the principles, methodologies and algorithms for synergistic knowledge discovery across multiple aligned information sources, and evaluates the corresponding benefits. Great challenges exist in "**Broad Learning**" for the effective fusion of relevant knowledge across different aligned information sources depends upon not only the relatedness of these information sources, but also the target application problems. "**Broad Learning**" aims at developing general methodologies, which will be shown to work for a diverse set of applications, while the specific parameter settings can be learned for each application from the training data.

Information transferred from other mature sources can help overcome the information sparsity and improve the recommendation results promisingly. As shown in Fig. 1(a), the online data source $\mathcal{G}^{(1)}$ is a newly created network source, in which the user-item rating information is not sufficient. However, by connecting $\mathcal{G}^{(1)}$ with a well-developed network source $\mathcal{G}^{(2)}$ which is full of user feedback information via the anchor links, we can extract some useful information from the user feedback information in $\mathcal{G}^{(2)}$, and transfer it to help the recommendation task in $\mathcal{G}^{(1)}$.

Another case is about the cross-source collaborative recommendation, where information can be transferred between different sources to help refine the recommendation results. As shown in Fig. 1(b), both $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$ are not new network sources, and each of them newly imports some items which have sufficient user feedbacks from the other data source. So in each of these data sources, the user feedback information of the newly imported items may not be sufficient for them to be recommend to the right users. However, by connecting $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$ together via anchor links, each of them may acquire some useful information of its newly imported items from the other. Thus the recommendation tasks of $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$ can collaborate together to achieve better results.

Cross-source recommendation via anchor links is still a new problem so far, and very few works have been done on this topic [13, 23, 24]. When dealing with the applications in Fig. 1, there are some new challenges that remain to be solved:

[1]www.imdb.com

- *Collaborative tasks of recommendation*: most of the existing transfer learning methods based on multi-source information aim at improving only one recommendation task in a given data source [13, 23, 24]. However, when two data sources have their own recommendation tasks, how to apply anchor links to make these recommendation tasks collaborate together, in this way each one's performances can be improved by the other one simultaneously (as shown in Fig. 1(b)), is still a new challenge.
- *Domain differences among networks from different sources*: most of the transfer learning methods for recommendation are designed to transfer information between data sets in the same network source (e.g., the location data set and the user relation data set of foursquare) [11, 12]. However, on the transferring of information among different network sources, things can be much different. For example, the language used in Douban[2] is mainly Chinese, but the language used in IMDb is mainly English. Thus when transferring the movie information between these networks, we should explore a proper way to deal with the language difference. So the question of "how to transfer" should be reconsidered.
- *Effective transfer of new information*: if we choose item anchor links to connect networks from different sources for recommendation, the transferred information can also be different from the information transferred by the existing methods which are not based on item anchor links [23, 24]. So it is important to make sure the transferred information is useful and is proper to be transferred by these links. Meanwhile, it's also important to explore the way of extracting the transferred information.

To overcome the above challenges, this paper proposes a *Cross-network Collaborative Matrix Factorization (CCMF)* framework based on the broad learning setting, which uses the anchor links between items to connect two networks from different sources for recommendation. Unlike user information which is usually anoymized, online item information is widely accessible, user can get item anchor links easily by directly matching their important information. For example, we can get movie anchor links by directly matching the movies' names, directors, and actors. Since item anchor links are much easier to obtain than user anchor links, the anchor link acquisition is no longer a challenge to our method. Besides, this paper explores novel ways to transfer the information, which directly relates to the items and is crucial to the recommendation with high information sparsity. The information includes the item similarity information and item latent information. And different from most of the traditional works, our approach can make the recommendation tasks in these two networks collaborate together via the information transfer process. At the same time, in order to overcome the cross-network domain differences, a domain adaptation matrix H is applied to modify the transfer of item latent information. Finally, this paper conducts different experiments to compare the performances of *CCMF* with several famous recommendation methods. The experimental results show that *CCMF* outperforms all of the compared methods, and prove that by using *CCMF*, the recommendation tasks in two networks can truly cooperate together and achieve better results.

[2]www.douban.com

The remaining part of this paper is organized as follows: At first, we introduce several important concepts and define our research problem in Section 2. The framework of *CCMF* is presented in detail in Section 3, and evaluated in Section 4. Finally, we briefly review the related works in Section 5 and conclude the paper in Section 6.

2 PROBLEM FORMULATION

Supposing that there are two datasets $\mathcal{G}^{(1)} = (\mathcal{V}^{(1)}, \mathcal{E}^{(1)})$ and $\mathcal{G}^{(2)} = (\mathcal{V}^{(2)}, \mathcal{E}^{(2)})$, which are collected from two different network sources respectively. The set of entities in $\mathcal{G}^{(1)}$ contains two kinds of entities, and can be represented as $\mathcal{V}^{(1)} = \mathcal{U}^{(1)} \cup \mathcal{I}^{(1)}$. $\mathcal{U}^{(1)} = \{u_0^{(1)}, u_1^{(1)}, ..., u_{a-1}^{(1)}\}$ is the set of user entities in $\mathcal{G}^{(1)}$. $\mathcal{I}^{(1)} = \{v_0^{(1)}, v_1^{(1)}, ..., v_{b-1}^{(1)}\}$ is the set of item entities in $\mathcal{G}^{(1)}$. Here, $|\mathcal{U}^{(1)}| = a$ and $|\mathcal{I}^{(1)}| = b$. $\mathcal{E}^{(1)} \subseteq \mathcal{U}^{(1)} \times \mathcal{I}^{(1)}$ is the set of ratings between user entities and item entities in $\mathcal{G}^{(1)}$. We define $\mathcal{G}^{(2)}$ in a similar way, where $|\mathcal{U}^{(2)}| = c$ and $|\mathcal{I}^{(2)}| = d$.

Since each user may assign a group of rating values to a group of item entities, we can create the user-item rating matrices $R^{(1)}$ and $R^{(2)}$ according to $\mathcal{E}^{(1)}$ and $\mathcal{E}^{(2)}$. Here $R^{(1)} = [R_{i,j}^{(1)}]_{a \times b}$ is an $a \times b$ matrix that refers to a users' ratings on b item entities in $\mathcal{G}^{(1)}$, where $R_{i,j}^{(1)}$ denotes user $u_i^{(1)}$'s rating value on the item entity $v_j^{(1)}$ in $\mathcal{G}^{(1)}$. Similarly, $R^{(2)} = [R_{i,j}^{(2)}]_{c \times d}$ is a $c \times d$ matrix that refers c users' ratings on d item entities in $\mathcal{G}^{(2)}$. Here, $R_{i,j}^{(1)}$ and $R_{i,j}^{(2)} \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, ?\}$, where the question mark "?" denotes a missing (unobserved) rating value.

Supposing that $\mathcal{A} \subseteq \mathcal{I}^{(1)} \times \mathcal{I}^{(2)}$ is the set of anchor links, which connect item entities in $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$. Thus there must exist an anchor link mapping function $\phi : \mathcal{A} \rightarrow T$, where matrix $T = [T_{i,j}]_{b \times d}$. Supposing that $l(v_i^{(1)}, v_j^{(2)}) \in \mathcal{A}$ is an anchor link that connects the $v_i^{(1)} \in \mathcal{I}^{(1)}$ with $v_j^{(2)} \in \mathcal{I}^{(2)}$. Thus if $l(v_i^{(1)}, v_j^{(2)})$ exists, $T_{i,j}$ is set to 1; otherwise, it is set to 0. Since all of the anchor links in $\mathcal{A}$ follow the one-to-one constraint, for all of the $T_{i,j} \in T$, we have: $\forall i, \forall j \left(\sum_{k=0}^{d} T_{i,k} \leq 1, \sum_{k=0}^{b} T_{k,j} \leq 1 \right)$.

Our goal is to predict the missing rating values in $R^{(1)}$ and $R^{(2)}$ by transferring the item information between $R^{(1)}$ and $R^{(2)}$ via T.

3 THE CCMF METHOD

In this section, we will introduce the *CCMF* method which connects two networks from difference sources for recommendation. We firstly review the basic *Low-Rank Matrix Factorization* framework, and then introduce the improved *CCMF* model by illustrating the way of extracting and transferring the information between two networks, as well as the way of applying the transferred information to improve the recommendation process.

3.1 Low-Rank Matrix Factorization

The *Low-Rank Matrix Factorization* [18] has been widely studied in recommendation systems. The basic idea of it is to factorize the user-item rating matrix R into two matrices U and V, representing user

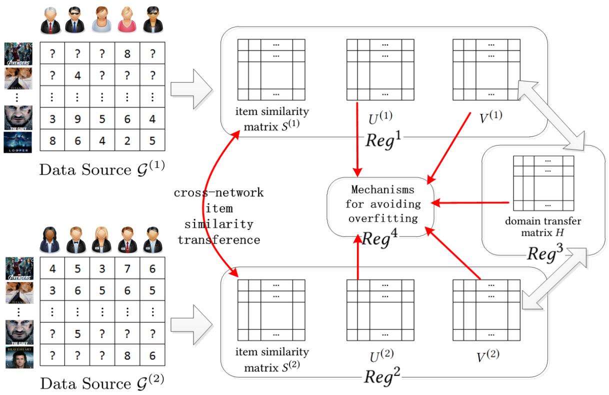


Figure 2: The main framework of *CCMF* method.

and item distributions on latent semantic, respectively. Then, the rating prediction can be made through these two specific matrices. This approach mainly minimizes the objective function:

$$\min_{U,V} L = \frac{1}{2}\sum_{i,j} W_{i,j}\left(R_{i,j} - U_i {V_j}^T\right)^2 + \frac{\lambda}{2}\left(\|U\|^2 + \|V\|^2\right) \tag{1}$$

where $W = [W_{i,j}]$ is a corresponding nonnegative weight matrix, if user i has rated item j, then $W_{i,j} = 1$; otherwise, $W_{i,j} = 0$. For a given matrix $\mathcal{R}$ ($\mathcal{R}$ can be U, V, W, R, etc.), $\mathcal{R}_k$ represents the row vector derived from the kth row of $\mathbb{R}$. $\frac{\lambda}{2}\left(\|U\|^2 + \|V\|^2\right)$ is the quadratic regularization term which aims to avoid overfitting, while λ represents the regularization parameter that is used to adjust the importance of the quadratic regularization term.

3.2 Cross-network Collaborative Matrix Factorization

The *Cross-network Collaborative Matrix Factorization(CCMF)* method considers the regularization on the information of two different network datasets $\mathcal{G}^{(1)}$, $\mathcal{G}^{(2)}$ and the anchor link based information transfer between them. The main framework of *CCMF* is illustrated in Fig. 2. Here, Reg^1 and Reg^2 represent the regularization term on intra-network information in $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$ respectively; Reg^3 represents the regularization term on the transfer of item latent factors between $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$; and Reg^4 represents the quadratic regularization term which aims to avoid overfitting. Then the optimization problem is to minimize the objective function $\mathcal{L}$, which calculates the sum of Reg^1, Reg^2, Reg^3 and Reg^4. The details of this *CCMF* framework will be presented in the following part of this subsection.

3.2.1 The regularization terms on intra-network information. According to Fig. 2, in Reg^1 we factorize the $\mathcal{G}^{(1)}$'s user-item rating matrix $R^{(1)} \in \mathbb{R}^{a \times b}$ into $U^{(1)} \in \mathbb{R}^{a \times m}$ and $V^{(1)} \in \mathbb{R}^{b \times m}$. Here m is the dimension number of latent factors in $U^{(1)}$ and $V^{(1)}$, constant a (constant b) denotes the number of users (items) in $\mathcal{G}^{(1)}$, and $m \ll \min(a, b)$. Similarly, in Reg^2, $R^{(2)} \in \mathbb{R}^{c \times d}$ is factorized into $U^{(2)} \in \mathbb{R}^{c \times n}$ and $V^{(2)} \in \mathbb{R}^{d \times n}$, where $n \ll \min(c, d)$.

Intuitively, similar items have similar features and are easy to get similar ratings by the same user. That is, the latent factor of an item is similar to the latent factors of items which are very similar to the item. Based on this assumption, for the regulation term Reg^k (where $k \in \{1, 2\}$), the *Item Similarity Regularization* is added to the basic *Low-Rank Matrix Factorization* framework as follows [25]:

$$Reg^k = \frac{1}{2}\sum_{i=0}^{a}\sum_{j=0}^{b} W_{i,j}^{(k)}\left(R_{i,j}^{(k)} - U_i^{(k)} V_j^{(k)T}\right)^2 + \frac{\gamma}{2}\sum_{i=0}^{b}\sum_{j=0}^{b} S_{i,j}^{(k)}\|V_i^{(k)} - V_j^{(k)}\|^2 \tag{2}$$

where $S^{(k)} = \{S_{i,j}^{(k)}\}$ is a corresponding item similarity matrix of $\mathcal{G}^{(k)}$, $S_{i,j}^{(k)}$ denotes the computed similarity between item i and item j in $\mathcal{G}^{(k)}$. $\sum_{i,j} S_{i,j}^{(k)}\|V_i^{(k)} - V_j^{(k)}\|^2$ enforces a large $S_{i,j}^{(k)}$ to have a small distance between $V_i^{(k)}$ and $V_j^{(k)}$ (i.e., similar users have smaller distance on latent factors). And γ represents the regularization parameter that is used to adjust the importance of *Item Similarity Regularization*.

There are many ways to compute the item similarities, some even use heterogeneous information [15, 25]. However, in this paper, we focus on presenting a basic framework which recommends items only by the information in the user rating matrices. So for a given network, its item similarity matrix is computed from its user rating matrix, which is as follows:

$$S = \left\{ S_{i,j} \middle| S_{i,j} = \frac{M_i {M_j}^T}{\|M_i\|\|M_j\|} \right\} \tag{3}$$

where $M = W^T * R^T$ and $*$ denotes element-wise multiplication.

3.2.2 Cross-network item similarity transfer. To any two item entities in two different networks that are aligned by an anchor link, although they represent the same item, their rating values can be very different. So when using Eq. (3) to compute the item similarities in $G^{(1)}$ and $G^{(2)}$ separately, two given items can also have different similarity values in different network sources. However, in real world, whether two items are similar or not is decided by their inherent characteristics, so their similarity should not change with networks. Besides, if the user-item rating information of two given item entities in one network is very sparse, their similarity computed from these rating information is inadequate to represent the similarity between their represented items in real world. So when computing the item similarities, we should utilize as more rating information as possible.

In our problem, a simple method to utilize more information for item similarity computation is to combine the user-item rating matrices $R^{(1)}$ and $R^{(2)}$ directly, and use the combined matrix to calculate the item similarities. However, this method only applies to the circumstance, where for any k, the entities $v_k^{(1)} \in G^{(1)}$ and $v_k^{(2)} \in G^{(2)}$ belong to the same item (i.e., for any k, all $R_{i,k}^{(1)}$ and $R_{j,k}^{(2)}$ are the ratings of the same item). In real world, an item entity in $G^{(1)}$ may not be aligned with any entity in $G^{(2)}$ via the anchor link. Moreover, when $i \neq j$, the ith item entity in $G^{(1)}$ and the jth item entity in $G^{(2)}$ may also represent the same item. So by directly combining the $R^{(1)}$ and $R^{(2)}$, the rating values of some other items may be misused to compute the similarity between two given items.

In the following part of this subsection, a novel method to transfer the item similarity information between $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$ is proposed. By using this method, the user-item ratings values in different networks can be correctly used to compute the item similarities, and at the same time, the recommendation tasks in these two networks can share the same item similarities.

Firstly, our method evaluates the amount of information used to computed the similarity of each pair of item entities in $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$ as follows:

$$Q_{i,j}^{(1)} = \sum_k W_{k,i}^{(1)} W_{k,j}^{(1)}, \; Q_{i,j}^{(2)} = \sum_k W_{k,i}^{(2)} W_{k,j}^{(2)} \tag{4}$$

where $Q_{i,j}^{(1)}$ ($Q_{i,j}^{(2)}$) is used to denote the amounts of the rating information, which is used to compute the similarities of the ith and jth item entities in $\mathcal{G}^{(1)}$ ($\mathcal{G}^{(2)}$), respectively. Thus we can get $Q^{(1)} = \{Q_{i,j}^{(1)}\} = W^{(1)T} W^{(1)}$ and $Q^{(2)} = \{Q_{i,j}^{(2)}\} = W^{(2)T} W^{(2)}$.

However, entity $v_i^{(1)} \in \mathcal{G}^{(1)}$ and entity $v_i^{(2)} \in \mathcal{G}^{(2)}$ may not represent the same item. So the matrix T, which is generated from all the anchor links between $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(1)}$, should be applied to transfer the similarity information. As a result, after transferring $Q^{(1)}$ to $\mathcal{G}^{(2)}$, we get the transferred matrix $\widetilde{Q}^{(1)} = T^T Q^{(1)} T$. Similarly, after transferring $Q^{(2)}$ to $\mathcal{G}^{(1)}$, we get $\widetilde{Q}^{(2)} = T Q^{(2)} T^T$. Finally, the way of computing $S_{i,j}^{(1)}$ is as follows:

$$S_{i,j}^{(1)} = \begin{cases} \dfrac{M_i^{(1)} M_j^{(1)T}}{\|M_i^{(1)}\| \|M_j^{(1)}\|}, \text{ if } Q_{i,j}^{(1)} > \widetilde{Q}_{i,j}^{(2)} \\ \dfrac{\widetilde{M}_i^{(2)} \widetilde{M}_j^{(2)T}}{\|\widetilde{M}_i^{(2)}\| \|\widetilde{M}_j^{(2)}\|}, \text{ if } Q_{i,j}^{(1)} < \widetilde{Q}_{i,j}^{(2)} \\ \dfrac{M_i^{(1)} M_j^{(1)T} + \widetilde{M}_i^{(2)} \widetilde{M}_j^{(2)T}}{(\|M_i^{(1)}\| + \|\widetilde{M}_i^{(2)}\|)(\|M_j^{(1)}\| + \|\widetilde{M}_j^{(2)}\|)}, \text{ otherwise} \end{cases} \tag{5}$$

where $M^{(1)} = (W^{(1)} * R^{(1)})^T$, $\widetilde{M}^{(2)} = T^T (W^{(2)} * R^{(2)})^T$. And $Q_{i,j}^{(1)} > \widetilde{Q}_{i,j}^{(2)}$ means $\mathcal{R}^{(1)}$ contains more rating information that can be used to compute $S_{i,j}^{(1)}$ than $\mathcal{R}^{(2)}$, so $S_{i,j}^{(1)}$ should be computed by the information in $\mathcal{R}^{(1)}$. For better understanding, we give an example: suppose that $T = [1,0,0;0,1,0;0,0,0]$, which means $v_0^{(1)}$ ($v_1^{(1)}$) and $v_0^{(2)}$ ($v_1^{(2)}$) represent the same item, but $v_2^{(1)}$ and $v_2^{(2)}$ represent different items. And suppose $R^{(1)} = [2,?,3;2,3,4;?,?,1]^T$, $R^{(2)} = [4,2,?;?,2,?;3,4,5]^T$; then we can get $W^{(1)} = [1,0,1;1,1,1;0,0,1]^T$, and $W^{(2)} = [1,1,0;0,1,0;1,1,1]^T$ from $R^{(1)}$ and $R^{(2)}$. According to Equations (3) and (4), the ratings in $R^{(1)}$ that can be used to compute $S_{0,1}^{(1)}$ are $R_{0,0}^{(1)}$, $R_{2,0}^{(1)}$, $R_{0,1}^{(1)}$ and $R_{2,1}^{(1)}$, $Q_{0,1}^{(1)} = [1,0,1][1,1,1]^T = 2$. Similarly, the ratings in $R^{(2)}$ that can be used to compute $S_{0,1}^{(2)}$ are $R_{1,0}^{(2)}$ and $R_{1,1}^{(1)}$, $Q_{0,1}^{(2)} = [1,1,0][0,1,0]^T = 1$. Then we can prove $Q_{0,1}^{(1)} > \widetilde{Q}_{0,1}^{(1)} = Q_{0,1}^{(2)}$, which is consistent with the fact that the ratings in $R^{(1)}$ that can be used to compute $S_{0,1}^{(1)}$ are more than the ratings in $R^{(2)}$ that can be used to compute $S_{0,1}^{(2)}$. So instead of transferring the information in $R^{(2)}$ to compute $S_{0,1}^{(1)}$, we use the information in $R^{(1)}$ to compute it. Besides, since $T_2 = [0,0,0]$, we can compute that $\widetilde{M}_2^{(2)} = 0$, and $Q_{k,3}^{(1)} = Q_{k,3}^{(1)} \geq \widetilde{Q}_{k,3}^{(2)} = \widetilde{Q}_{3,k}^{(2)} = 0$ where $k \in \{0,1,2\}$. As a result, although $v_2^{(1)}$'s rating information is very sparse, the sufficient rating information of $v_2^{(2)}$ will not be misused to compute $v_2^{(1)}$'s similarities to the other entities in $\mathcal{G}^{(1)}$.

Similarly, after setting $M^{(2)} = W^{(2)} * R^{(2)}$, $\widetilde{M}^{(1)} = (W^{(1)} * R^{(1)})T$, we can compute $S_{i,j}^{(2)}$ by replacing the superscript (1) with (2) and replacing (2) with (1) in Eq. (5) simultaneously.

3.2.3 Cross-network item latent factor transfer. Since if there exists an anchor link that connects $v_i^{(1)} \in \mathcal{G}^{(1)}$ and $v_j^{(2)} \in \mathcal{G}^{(1)}$, then $T_{i,j} = 1$, $v_i^{(1)}$ and $v_j^{(2)}$ must represent the same item. Suppose that $V^{(1)}$ and $V^{(2)}$ are in the same domain, the latent factor vectors $V_i^{(1)}$ and $V_j^{(2)}$ should also be the same. However, because some items only have entities in $\mathcal{G}^{(1)}$ or $\mathcal{G}^{(2)}$ (i.e., the row dimensions of $V^{(1)}$ and $V^{(2)}$ can be different), we can't directly set $V^{(1)} = V^{(2)}$. Instead, we set $T^T V^{(1)} = T^T T V^{(2)}$, where T is used to ensure only the two latent factor vectors of the same item are restricted to be the same. We also notice that although $V^{(1)}$ and $V^{(2)}$ are in the same domain, the latent user tastes and item factors in $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$ can still be a bit different due to each network's specific contexture, e.g., advertisements or promotions on the service provider's website [13]. So we relax this requirement and only require $T^T V^{(1)}$ and $T^T T V^{(2)}$ to be similar, i.e., require $\|T^T V^{(1)} - T^T T V^{(2)}\|^2$ to be as small as possible. Moreover, in most case $V^{(1)}$ and $V^{(2)}$ are in different domains. Due to the domain differences (e.g., different user cultures, different user tastes, and different languages), a given item may have different latent factors in different networks, and the column dimensions of $V^{(1)}$ and $V^{(2)}$ can also be different (i.e., m and n can have different values). So we apply an item latent domain adaptation matrix $H \in \mathbb{R}^{m \times n}$ to bridge the domain differences between $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$, in this way to make $T^T V^{(1)} H$ and $T^T T V^{(2)}$ be as similar as possible. Finally, Reg^3 is used to represent the regularization term on the transfer of item latent factors between $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$ as follows:

$$Reg^3 = \frac{1}{2} \|T^T V^{(1)} H - T^T T V^{(2)}\|^2 \tag{6}$$

By minimizing Reg^3, the latent information of the items owned by both $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$ will be transfered between $V^{(1)}$ and $V^{(2)}$ according to the anchor link information in T.

3.2.4 A Unified Collaborative Regularization. In Reg^1, Reg^2 and Reg^3 there are five matrices remain to be computed, including $V^{(1)}$, $V^{(2)}$, $U^{(1)}$, $U^{(2)}$ and H. In order to avoid overfitting when computing them, a quadratic regularization term Reg^4 is created as follows:

$$Reg^4 = \frac{\lambda_1}{2}\left(\|U^{(1)}\|^2 + \|V^{(1)}\|^2\right) + \frac{\lambda_2}{2}\left(\|U^{(2)}\|^2 + \|V^{(2)}\|^2\right) + \frac{\lambda_3}{2}\|H\|^2 \tag{7}$$

where λ_1, λ_2 and λ_3 are the regularization parameters. As a result, the optimization function of our *CCMF* method is defined as follows:

$$\min_{U^{(1)}, V^{(1)}, U^{(2)}, V^{(2)}, H} \mathcal{L} = Reg^1 + \alpha Reg^2 + \beta Reg^3 + Reg^4 \tag{8}$$

where α and β are the regularization parameters which relate to the importance of Reg^2 and Reg^3 respectively. And if we think Reg^2 is more important than Reg^3, we can assign α a value which is bigger than β.

3.2.5 Optimization Algorithm. Since the objective function in Eq. (8) is non-convex, we adopt an iterative optimization algorithm that alternatively optimizes each variable while fixing others until convergence. Specifically, by calculating the partial derivatives of the objective $\mathcal{L}$ with respect to $U^{(1)}$, $V^{(1)}$, $U^{(2)}$, $V^{(2)}$ and H, respectively, and setting them to 0, we have:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial U^{(1)}} = \left(W^{(1)} * (U^{(1)}V^{(1)T} - R^{(1)})\right) V^{(1)} + \lambda_1 U^{(1)} = 0 \\ \frac{\partial \mathcal{L}}{\partial V^{(1)}} = \left(W^{(1)T} * (V^{(1)}U^{(1)T} - R^{(1)T})\right) U^{(1)} + \lambda_1 V^{(1)} + 2\gamma L^{(1)} V^{(1)} \\ \qquad + \beta T \left(T^T V^{(1)} H - T^T T V^{(2)}\right) H^T = 0 \\ \frac{\partial \mathcal{L}}{\partial U^{(2)}} = \alpha \left(W^{(2)} * (U^{(2)}V^{(2)T} - R^{(2)})\right) V^{(2)} + \lambda_2 U^{(2)} = 0 \\ \frac{\partial \mathcal{L}}{\partial V^{(2)}} = \alpha \left(W^{(2)T} * (V^{(2)}U^{(2)T} - R^{(1)T})\right) U^{(2)} + \lambda_2 V^{(2)} + 2\alpha\gamma L^{(2)} V^{(2)} \\ \qquad - \beta T^T T \left(T^T V^{(1)} H - T^T T V^{(2)}\right) = 0 \\ \frac{\partial \mathcal{L}}{\partial H} = \beta V^{(1)T} T \left(T^T V^{(1)} H - T^T T V^{(2)}\right) + \lambda_3 H = 0 \end{cases} \tag{9}$$

where $L^{(k)} = D^{(k)} - S^{(k)}, k \in \{1, 2\}$ and $D^{(k)}$ is a diagonal matrix with elements $D^{(k)}_{i,i} = \sum_j S^{(k)}_{i,j}$.

By transforming all the equations in Eq. (9) into their corresponding linear system forms, we can solve them directly. For example, supposing that the vec operator reshapes a matrix $A = [a_1, a_2, ..., a_n]$ to its column vector form $vec(A) = [a_1^T, a_2^T, ..., a_n^T]^T$ by stacking the column vectors of A below one another. And using $vec(ABC^T) = (C \otimes A)vec(B)$, where A, B and C are three arbitrary matrices, $\otimes$ is the *Kronecker product*. We can rewrite $\frac{\partial \mathcal{L}}{\partial U^{(1)}} = 0$ as a linear system:

$$AX = B \tag{10}$$

where $A = \left(V^{(1)T} \otimes I_1\right) diag\left(vec(W^{(1)})\right)(V^{(1)} \otimes I_1) + \lambda_1, X = vec(U^{(1)})$, $B = \left(V^{(1)T} \otimes I_1\right) diag\left(vec(W^{(1)})\right) vec(R^{(1)})$, and I_1 is an $a \times a$ identity matrix.

Then, since A is invertible, we have the solution in the vector form as $vec(U^{(1)}) = A^{-1}B$. Thus $U^{(1)}$ can be updated according to the value of $A^{(-1)}B$. Similarly, we can update $V^{(1)}$, $U^{(1)}$, $V^{(1)}$ and H by solving their corresponding $AX = B$ forms in the same way. However, for each of these five matrix variables, the computation of its related A^{-1} is usually time consuming. Alternatively, we can solve Eq. (9) iteratively by using the conjugate gradient(CG) method [4], which only needs to perform matrix multiplications on these equations in Eq. (9) respectively without rewriting them to their linear system forms. In this way, the explicit representations of matrix A^{-1} for all the five matrix variables are not needed.

The whole procedure of *CCMF* is summarized in Algorithm 1.

4 EXPERIMENT

To verify the superiority of our approach, in this section we conduct several experiments to compare the proposed *CCMF* approach with five baseline recommendation methods.

4.1 Data Preparation

In this paper, we crawl two datasets $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$. $\mathcal{G}^{(1)}$ is from Douban Movie[3]. Douban is a Chinese SNS website allowing registered users to record information and create contents related to film, books, music, and recent events and activities in Chinese cities. As one of the most successful service branch of Douban, Douban Movie provides comprehensive knowledge about recent and past movies across the world together with the user reviews. $\mathcal{G}^{(2)}$ is from IMDb (short for the Internet Movie Database), which is owned by Amazon.com, and is an international online database of information related to films, television programs and video games. The anchor links for our experiment are the inter-network links which connect the movie entities across $\mathcal{I}^{(1)} \subset \mathcal{G}^{(1)}$ and $\mathcal{I}^{(2)} \subset \mathcal{G}^{(2)}$. These links are crawled by tracing the property of *IMDb Link* on the homepages of each movies in Douban.com.

In our experiments, for better analyzing the effect of applying anchor links to the recommendation process, we only focus on the movie items shared by $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$. So we select 800 movies, each of which has its entities in $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$ simultaneously. Then we use their entities in $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$ to form the experimental item entities sets $\mathcal{I}^{(1)} \subset \mathcal{G}^{(1)}$ and $\mathcal{I}^{(2)} \subset \mathcal{G}^{(2)}$ respectively. Since each movie entity in $\mathcal{I}^{(1)}$ is connected with a movie entity in $\mathcal{I}^{(2)}$ via an anchor link, we can form the corresponding movie anchor link set $\mathcal{A}$ for $\mathcal{I}^{(1)}$ and $\mathcal{I}^{(2)}$. And then in each network, we randomly select 800 users who have rated some of these 800 movies, and form the experimental user sets $\mathcal{U}^{(1)} \subset \mathcal{G}^{(1)}$ and $\mathcal{U}^{(2)} \subset \mathcal{G}^{(2)}$ respectively. Finally, we can generate the user rating set $\mathcal{E}^{(k)}$ according to $\mathcal{I}^{(k)}$ and $\mathcal{U}^{(k)}$ as well as the related ratings, and create the original user rating matrix $\dot{R}^{(k)}$ according to $\mathcal{E}^{(k)}$, where $k \in 1, 2$. However, since the

Algorithm 1 Algorithm framework of *CCMF*

Input: $\mathcal{G}^{(1)}$, $\mathcal{G}^{(2)}$: two heterogeneous information networks; $\mathcal{A}$: the anchor link set; α, β, γ, λ_1, λ_2, λ_3:controlling parameters defined above; ϕ: the anchor link mapping function;
Output: $U^{(1)}$ and $U^{(2)}$: the users' latent semantic distribution matrices; $V^{(1)}$ and $V^{(2)}$: The items' latent semantic distribution matrices;
Create the user-item rating matrices $R^{(1)}$ and $R^{(2)}$ according to the rating information in $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$
Use ϕ to create matrix T according to $\mathcal{A}$
Create the weight matrices $W^{(1)}$, $W^{(2)}$ according to the rating information in $\mathcal{G}^{(1)}$, $\mathcal{G}^{(2)}$
According to the method in Section 3.2.2, compute $S^{(1)}$, $S^{(2)}$ by $W^{(1)}$, $W^{(2)}$, $R^{(1)}$ and $R^{(2)}$
Calculate $L^{(1)}$, $L^{(2)}$ by $S^{(1)}$ and $S^{(2)}$
Initialize $U^{(1)}$, $V^{(1)}$, $U^{(2)}$, $V^{(2)}$ and H
repeat
    Update $U^{(1)}$ by solving $\frac{\partial \mathcal{L}}{\partial U^{(1)}} = 0$ in Eq. (9)
    Update $U^{(2)}$ by solving $\frac{\partial \mathcal{L}}{\partial U^{(2)}} = 0$ in Eq. (9)
    Update $V^{(1)}$ by solving $\frac{\partial \mathcal{L}}{\partial V^{(1)}} = 0$ in Eq. (9)
    Update $V^{(2)}$ by solving $\frac{\partial \mathcal{L}}{\partial V^{(2)}} = 0$ in Eq. (9)
    Update H by solving $\frac{\partial \mathcal{L}}{H} = 0$ in Eq. (9)
until Eq. (8 converges)

Table 1: Statistics of the Datesets

	property	network dataset $\mathcal{G}^{(1)}$	network dataset $\mathcal{G}^{(2)}$
# entity	user	800	800
	movie	800	800
# relation	rating	66,226	84,394
	anchor link	800	800

[3]movie.douban.com

$\dot{R}^{(1)}_{i,j} \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, ?\}$ while $\dot{R}^{(2)}_{i,j} \in \{10, 20, 30, 40, 50, ?\}$, we set $R^{(1)} = \dot{R}^{(1)}$ and set $R^{(2)}_{i,j}$ as follows:

$$R^{(2)}_{i,j} = \begin{cases} \dot{R}^{(2)}_{i,j}/5 & \text{if } \dot{R}^{(2)}_{i,j} \neq ? \\ ? & \text{if } \dot{R}^{(2)}_{i,j} = ? \end{cases} \tag{11}$$

And $R^{(1)}$ and $R^{(2)}$ are finally used as the user rating matrices by the recommendation methods in our experiments.

The detailed statistics of the datasets are shown in Table 1.

4.2 Compared Methods

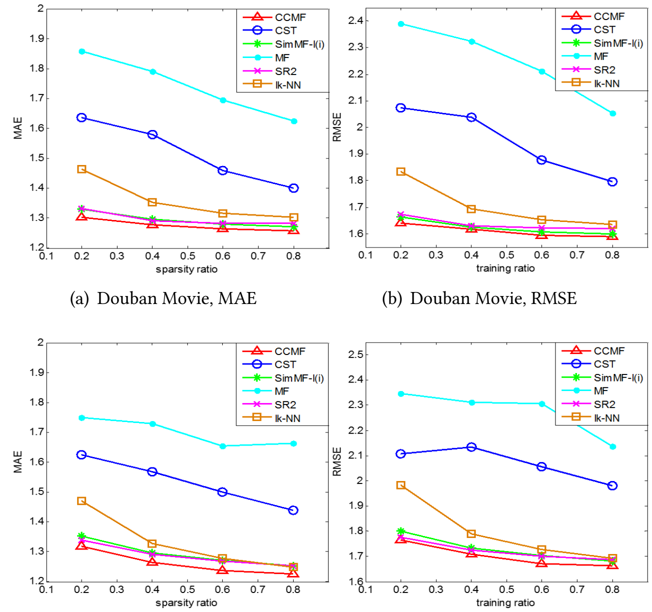
In order to demonstrate the effectiveness of the proposed *CCMF*, we compare it with five baseline methods. so in total, there are six methods to be compared. The compared methods are summarized as follows:

- *Item-based k-Nearest Neighbors Algorithm (Ik-NN)*: It is one of the most famous collaborative filtering methods, which recommends each item according to the rating information of its top-k nearest items [14]. According to the previous work [14], the *Adjusted Cosine Similarity* is chosen to compute the item similarities in this way to get better results.
- *Low-rank Matrix Factorization (LMF)*: It is proposed by Nathan Srebro and Tommi Jaakkola [18] and has been widely studied in many recommendation systems.
- *SimMF-I(i)*: It is a state-of-art matrix factorization based recommendation framework [15], which combines user-item ratings and item similarities for recommendation. For fair comparisons, we assume that other item relation information is unavailable, and the item similarities are computed from the rating information by Eq. (3).
- *SR2*: It is a matrix factorization based recommendation framework proposed by Ma et al. [10], which apply the user similarities to the recommendation process. According to the evaluations in [10], we choose the *PCC* [10] method to compute the user similarities in this way to get better results.
- *CST*: It is a matrix factorization based recommendation framework, which can transfer item latent factors across different networks. Different from our *CCMF* method, this method doesn't consider item similarities, and doesn't provide a way to bridge the domain differences between two networks [12].
- *CCMF*: This is our proposed *Cross-network Collabrative Matrix Factorization* method.

To make fair comparisons, for *LMF, SimMF-I(i), SR2, CST* and *CCMF*, we set the length of all latent factor vectors as 20 and set all the parameters used to avoid overfitting as 1.0. For *CCMF*, we also set the regularization parameter α as 1.0. And for other parameters, we do experiments to find their approximately optimal values for each method, and use these approximately optimal parameter values in the performance comparison experiments.

4.3 Evaluation Metrics

In order to evaluate the effectiveness of these compared methods, we use two different metrics, namely Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). Both of them are used to evaluate the total difference between the predicted user ratings and the user ratings in the test set. The metric MAE is defined as:



(a) Douban Movie, MAE (b) Douban Movie, RMSE

(c) IMDB, MAE (d) IMDB, RMSE

Figure 3: The performance comparisons on the item recommendation task in a given new network.

$$MAE = \frac{1}{N_t} \sum_{(i,j,k,R^{(k)}_{i,j}) \in T_E} |R^{(k)}_{i,j} - \hat{R}^{(k)}_{i,j}| \tag{12}$$

where $R^{(k)}_{i,j}$ is the actual rating value that user $u^{(k)}_i \in \mathcal{G}^{(k)}$ assigns to item $v^{(k)}_j \in \mathcal{G}^{(k)}$, and $\hat{R}^{(k)}_{i,j}$ denotes the predicted rating value that $u^{(k)}_i$ may assign to $v^{(k)}_j$. Particularly, $\hat{R}^{(k)}_{i,j}$ can be calculated by $U^{(k)}_i V^{(k)T}_j$ in our model. Moreover, T_E is the test set of user ratings, and N_t is the number of ratings in T_E.

RMSE is defined as:

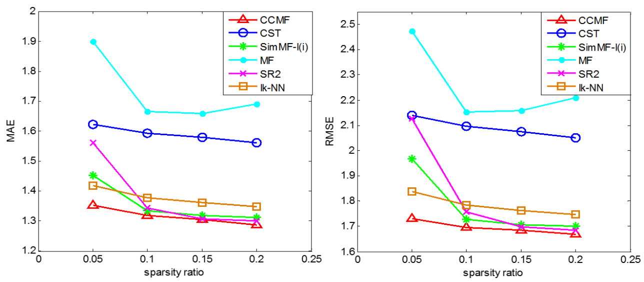
$$RMSE = \sqrt{\frac{1}{N_t} \sum_{(i,j,k,R^{(k)}_{i,j}) \in T_E} \left(R^{(k)}_{i,j} - \hat{R}^{(k)}_{i,j}\right)^2} \tag{13}$$

From the definitions, we can see that a smaller value of *MAE* or *RMSE* means a better performance.

4.4 Performance Comparisons on the Recommendation Task in a Given Network

In this subsection, we conduct experiments to compare the performances of the experimental methods in the circumstance shown in Fig. 1(a). In this circumstance, the movie recommendation should only be done in a new network $\mathcal{G}^{(a)}$, whose user feedback information is insufficient for its recommendation task to achieve good results. And the information in an old network $\mathcal{G}^{(b)}$ which has sufficient user feedback information can be transferred to help the recommendation task in $\mathcal{G}^{(a)}$.

So in each of these experiments, we set a training ratio r_t (r_t denotes the degree of information sparsity of $\mathcal{G}^{(a)}$, the smaller it is, the sparser $\mathcal{G}^{(a)}$'s information is.), and randomly sample $1 - r_t$ of



(a) MAE (b) RMSE

Figure 4: The performance comparisons in the circumstance where both of the aligned networks should recommend the newly imported items to their users.

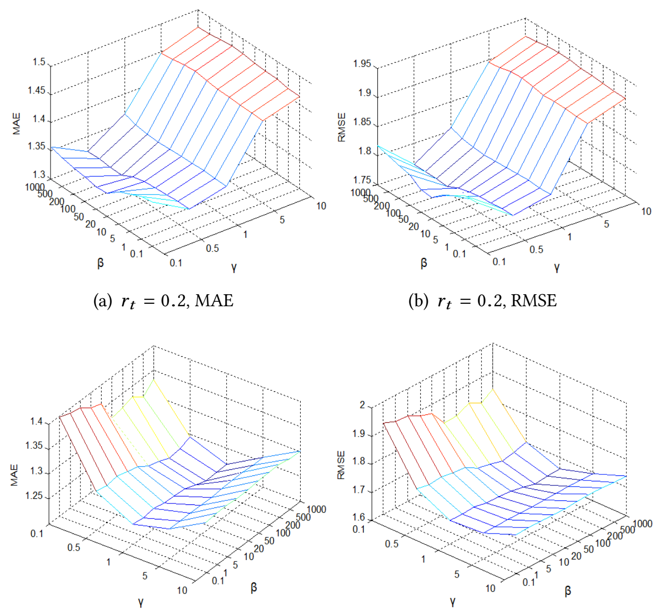
the observed ratings from the collected experimental data of $\mathcal{G}^{(a)}$, then use the sampled ratings to form the test set T_E. The rest of observed ratings in $\mathcal{G}^{(a)}$ and $\mathcal{G}^{(b)}$ are used to form the training set T_R. The value of r_t is selected from $\{0.2, 0.4, 0.6, 0.8\}$. If $a = 1$ then $b = 2$, and if $a = 2$ then $b = 1$. The random sampling was carried out 5 times independently, and the average results are shown in Fig. 3, in which the title of each subgraph is formed by the name of $\mathcal{G}^{(a)}$ and the metric.

From the results we can see that: 1) Our *CCMF* method outperforms all the baseline methods in this group of experiments, which suggests that it can effectively transfer information from an old network to help the recommendation task in a new network. 2) *CCMF* method outperforms the *SimMF-I(i)* method in different data sets, which proves the importance of applying anchor links to transfer information from an old network to help the recommendation task in a new network. Since without the techniques of information transfer via anchor links, *CCMF* will degrade to the *SimMF-I(i)* method. 3) *CCMF* method significantly outperforms the *CST* method, which means the domain adaptation matrix and item similarity transfer are important to the process of transferring information from an old network to help the recommendation task in a new network.

4.5 Performance Comparisons on the Recommendation Tasks in Multiple Networks

In this subsection, we conduct experiments to test the performances of the experimental methods on dealing with the circumstance shown in Fig. 1(b), where there are two networks, and each one of them newly brings in some item entities from the other network, thus these new entities should be recommended to some users as soon as possible. And in each network, the newly imported entities may not have enough user feedback information for recommendation, we can test whether our *CCMF* method can outperform other baseline methods by making these two networks’ recommendation tasks cooperate together.

So in these experiments, we firstly partition the collected movie anchor link set $\mathcal{A} = \{l(v_i^{(1)}, v_j^{(2)})\}$ with 5-fold cross validation: one fold as $\mathcal{A}_x$, the links in which connect the old movie entities in $\mathcal{G}^{(1)}$ with the newly imported movie entities in $\mathcal{G}^{(2)}$; one fold as $\mathcal{A}_y$, the links in which connect the old movie entities in $\mathcal{G}^{(2)}$ with the newly imported movie entities in $\mathcal{G}^{(1)}$; and the remaining 3



(a) $r_t = 0.2$, MAE (b) $r_t = 0.2$, RMSE

(c) $r_t = 0.8$, MAE (d) $r_t = 0.8$, RMSE

Figure 5: Performances of *CCMF* with varying β, γ and r_t on Douban Movie dataset. The lower, the better.

folds are formed by the anchor links among the old movies entities in $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$. Secondly, we create an empty entity set $\mathcal{I}_n$, and according to each link $l(v_i^{(1)}, v_j^{(2)}) \in \mathcal{A}_x$, we select the movie entity $v_i^{(1)} \in \mathcal{G}^{(1)}$ then add it to $\mathcal{I}_n$. Similarly, according to each $l(v_i^{(1)}, v_j^{(2)}) \in \mathcal{A}_y$, we select $v_j^{(2)} \in \mathcal{G}^{(2)}$ and add it to $\mathcal{I}_n$. In this way, $\mathcal{I}_n$ contains all of the newly imported movie entities in $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$. Thirdly, a sparsity ratio r_s is selected to denote the degree of rating information sparsity for all the newly imported items in $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$. And to all the observed ratings relate to the items in $\mathcal{I}_n$, r_s of them are randomly selected out to form the set T_{R1}, the rest of them are used to form the test set T_E. Then we randomly select 60% of the observed ratings, which are related to all the old movie entities in $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$ (i.e., which are not related to the items in $\mathcal{I}_n$), to form the set T_{R2}. Thus we get our training set $T_R = T_{R1} \cup T_{R2}$. To each network, since the imported items are very new, their ratings can be very sparse, as a result, the value of r_s is selected from $\{0.05, 0.10, 0.15, 0.2\}$.

The results are shown in Fig. 4, from which we can observe that: 1) By making the recommendation tasks in different networks collaborate together via anchor links, our *CCMF* method can outperform all of the other baseline methods on recommending the item entities in each network, which are newly imported from the other network. 2) Because *CCMF* provides an effective way for each network’s recommendation task to get useful information form the other network simultaneously, when the information of the newly imported items is very sparse, *CCMF* shows significant advantages over other base-line methods.

4.6 Parameter Study on β and γ

Since other kinds of parameters have been studied in previous matrix factorization methods [15], here we only do parameter study on β and γ, which relate to the two kinds of transferred information in our *CCMF* method. On one hand, if the user-item matrices for recommendation are factorized with a very small value of β and γ, *CCMF* will ignore the item similarities and item information transfer. On the other hand, if β and γ have very large values, the item similarity information and the process of item information transfer will dominate the model learning process. Intuitively, we need to set moderate values for β and γ to achieve good performances. As a result, we will analyze how the changes of β and γ effect the final recommendation accuracy in this section.

Choosing $\mathcal{G}^{(1)}$ as the network where the recommendation task is conducted, we firstly set the training ratio r_t as 0.2 or 0.8, and use $1 - r_t$ as the sample ratio to randomly sample the observed user-item ratings from $\mathcal{G}^{(2)}$, and then use the sampled ratings to form the test set T_E. The rest of observed ratings in $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$ are used to form the training set T_R. Here, $r_t = 0.2$ denotes that network $\mathcal{G}^{(1)}$ is very new and only contains a small amount of ratings, while $r_t = 0.8$ denotes that $\mathcal{G}^{(1)}$ is not very new and has a certain amount of ratings.

Figure 5 shows the impacts of β and γ on *MAE* and *RMSE* in *CCMF* model. We can find that with the same r_t, the performances of *CCMF* on MAE and RMSE have very similar trend. Moreover, the values of β and γ affect recommendation results significantly, which demonstrates that incorporating the multi-source item latent information and the item similarity information can greatly affect the recommendation accuracy. The results indicate that β and γ should be set with moderate values to make *CCMF* perform well: for $r_t = 0.2$, *CCMF* can achieve the best performance when $\beta = 100$ and $\gamma = 0.5$; while for $r_t = 0.8$, *CCMF* can achieve the best performance when $\beta = 200$ and $\gamma = 0.5$. And for very small β and γ, *CCMF* will degrade to two traditional *LMF* models, which make its MAE and RSME increase to higher and stable values (i.e., bad performance). For large β and γ, the item similarity information and the process of item information transfer will dominate model learning process, which also make the MAE and RSME values of *CCMF* increase.

5 RELATED WORKS

Anchor link prediction is an important research problem and several works have been published on this topic in the past seven years. Most of these works aim at connecting the user accounts of common users across different networks [5–8, 26, 27, 35], among them: S. Liu et al. [7] propose a framework to connect user accounts across heterogeneous social media platforms by using multiple user features. Xiangnan Kong et al. [5] explore the way of extracting heterogeneous features from multiple heterogeneous networks for anchor link prediction. And Y. Zhang et al. [35] develop a general cross-network user alignment model which can support the integration of a number of networks.

In order to recommend to online users with the information entities that match their interests, a lot of recommendation methods have been proposed so far. Among them, collaborative filtering is one of the most popular techniques, which makes automatic predictions on the new interests of a user by the user-item rating values on his/her other interested items, or the rating values from the other similar users. Collaborative filtering methods can be classified into two types of approaches: memory-based method and model-based method [14]. Different from the collaborative filtering which directly uses the rating values, the low rank matrix factorization method firstly factorizes user-item rating matrix into two low rank user-specific and item-specific matrices, then utilizes the factorized matrices to make further recommendations [18].

However, in some networks, user-item rating information is usually very sparse, which makes many traditional recommendation methods cannot perform very well. In order to alleviate the information sparse problem, several methods have been proposed [3, 9, 15, 17, 22, 25] to integrate some auxiliary information besides the user-item rating information into the matrix factorization process, in this way to have sufficient information for recommendation. Among them, [17] aims at jointly modeling a relational database and an item-users rating matrix to improve collaborative filtering. Yu Xiao et al. [25] propose the way of extracting item similarities from multiple types of relation information and applying these similarities to the matrix factorization process, in this way to get sufficient information for recommendation. Shi Chuan et al. [15] propose a flexible regularization framework, which integrates different types of the user relation information and item relation information into the recommendation process. However, the auxiliary information utilized by these methods is contained in the same data source where the recommendation task is conducted (e.g., utilizing the user relations in a given data source to help recommending the items in the same data source [9]). And when this kind of auxiliary information is insufficient or unavailable in the data source, the information sparsity is still a big problem. For example, in the real world e-commerce sites (like Amazon), few social relations really exist among users, so the methods like [9], [15] and [3] which rely on social relations for better recommendation performances become powerless.

Nevertheless, some works also explore the way of transferring the information from the source network to the target network to alleviate the information sparse problem. Among them, methods like [2] and [16] try to recommend items to the users in a given network $\mathcal{G}^{(1)}$ according to the preferences of users in the other network $\mathcal{G}^{(2)}$. However, since the information transferred to $\mathcal{G}^{(1)}$ may not be related enough to the information in $\mathcal{G}^{(2)}$, these methods usually face the "negative transfer" problem [21], which often causes bad recommendation performances. Because anchor links can connect two networks from different sources together, via anchor links, the information which is closely related to both of these two networks can be transferred directly between them. But only a very few works have been done to explore the anchor link based recommendation methods. Ming Yan et al. [23, 24] explore the way of recommending videos for YouTube users by transferring users' social and content information from Twitter network via user anchor links, however, since user anchor links are usually very hard to collect due to the privacy concerns , their works can hardly adapt to other applications. Weike Pan et al. [13] propose a transfer learning framework which integrates multi-source network information for recommendation via the user and item anchor links, however, they never consider the item similarities, nor do they discuss the domain differences between different networks.

6 CONCLUSIONS

In this paper, we propose a *Cross-network Collaborative Matrix Factorization (CCMF)* framework to integrate multi-source information for recommendation tasks based on broad learning setting, in this way to solve the information sparse problem caused by different reasons. Basing on item anchor links, *CCMF* can transfer item similarity information and item latent information across networks from different sources. And different from most traditional recommendation methods, *CCMF* can make the recommendation tasks from different network sources collaborate together. During the information transfer process, a novel method is introduced to keep the consistency of item similarities between two different networks, and a domain adaptation matrix is used to overcome the domain difference problem. We conduct experiments to compare the proposed *CCMF* method with several widely used or state-of-the-art recommendation techniques, and the experimental results reflect that *CCMF* outperforms other methods in different circumstances. Since our *CCMF* is a basic framework which only exploits user-item rating information for recommendation, in future work we will study on how to properly integrate the knowledge of heterogeneous information into *CCMF* in this way to achieve better performances.

ACKNOWLEDGMENTS

This work is supported in part by National Key fundamental Research and Development Program of China (No.2013CB329601, No.2013CB329604, No.2013CB329606), National Natural Science Foundation of China (No. 61502517, No.61372191, No. 61572492, No.61503253). This work is also funded by the major pre-research project of National University of Defense Technology.

REFERENCES

[1] Indrajit Bhattacharya and Lise Getoor. 2007. Collective entity resolution in relational data. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1, 1 (2007), 5.

[2] Wei Chen, Wynne Hsu, and Mong Li Lee. 2013. Making recommendations from multiple domains. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 892–900.

[3] Meng Jiang, Peng Cui, Xumin Chen, Fei Wang, Wenwu Zhu, and Shiqiang Yang. 2015. Social Recommendation with Cross-Domain Transferable Knowledge. *IEEE Transactions on Knowledge & Data Engineering* 27, 11 (2015), 3084–3097.

[4] Carl T Kelley. 1999. *Iterative methods for optimization*. Vol. 18. Siam.

[5] Xiangnan Kong, Jiawei Zhang, and Philip S Yu. 2013. Inferring anchor links across multiple heterogeneous social networks. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. ACM, 179–188.

[6] Dong Liu and Quan Yuan Wu. 2013. Cross-Platform User Profile Matching in Online Social Networks. In *Applied Mechanics and Materials*, Vol. 380. Trans Tech Publ, 1955–1958.

[7] Siyuan Liu, Shuhui Wang, Feida Zhu, Jinbo Zhang, and Ramayya Krishnan. 2014. Hydra: Large-scale social identity linkage via heterogeneous behavior modeling. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM, 51–62.

[8] Chun-Ta Lu, Hong-Han Shuai, and Philip S Yu. 2014. Identifying your customers in social networks. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, 391–400.

[9] Chen Luo, Wei Pang, Zhe Wang, and Chenghua Lin. 2014. Hete-cf: Social-based collaborative filtering recommendation using heterogeneous relations. In *2014 IEEE International Conference on Data Mining*. IEEE, 917–922.

[10] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February*. 287–296.

[11] Weike Pan. 2016. A survey of transfer learning for collaborative recommendation with auxiliary data. *Neurocomputing* 177 (2016), 447–453.

[12] Weike Pan, Nathan N Liu, Evan W Xiang, and Qiang Yang. 2011. Transfer learning to predict missing ratings via heterogeneous user feedbacks. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, Vol. 22. 2318.

[13] Weike Pan, Evan Wei Xiang, Nathan Nan Liu, and Qiang Yang. 2010. Transfer Learning in Collaborative Filtering for Sparsity Reduction.. In *AAAI*, Vol. 10. 230–235.

[14] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. ACM, 285–295.

[15] Chuan Shi, Jian Liu, Fuzhen Zhuang, S Yu Philip, and Bin Wu. 2016. Integrating heterogeneous information via flexible regularization framework for recommendation. *Knowledge and Information Systems* (2016), 1–25.

[16] Yue Shi, Martha Larson, and Alan Hanjalic. 2011. Tags as bridges between domains:improving recommendation with tag-induced cross-domain collaborative filtering. In *User Modeling, Adaption and Personalization - International Conference, Umap 2011, Girona, Spain, July 11-15, 2011. Proceedings*. 305–316.

[17] Ajit P Singh and Geoffrey J Gordon. 2008. Relational learning via collective matrix factorization. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 650–658.

[18] Nathan Srebro, Tommi Jaakkola, and others. 2003. Weighted low-rank approximations. In *Icml*, Vol. 3. 720–727.

[19] Yizhou Sun, Rick Barber, Manish Gupta, Charu C Aggarwal, and Jiawei Han. 2011. Co-author relationship prediction in heterogeneous bibliographic networks. In *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*. IEEE, 121–128.

[20] Michail Tsikerdekis and Sherali Zeadally. 2014. Multiple account identity deception detection in social media using nonverbal behavior. *IEEE Transactions on Information Forensics and Security* 9, 8 (2014), 1311–1321.

[21] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. 2016. A survey of transfer learning. *Journal of Big Data* 3, 1 (2016), 1–40.

[22] Jian Wu, Liang Chen, Qi Yu, Panpan Han, and Zhaohui Wu. 2015. Trust-aware media recommendation in heterogeneous social networks. *World Wide Web* 18, 1 (2015), 139–157.

[23] Ming Yan, Jitao Sang, and Changsheng Xu. 2015. Unified youtube video recommendation via cross-network collaboration. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*. ACM, 19–26.

[24] Ming Yan, Jitao Sang, Changsheng Xu, and M Shamim Hossain. 2016. A Unified Video Recommendation by Cross-Network User Modeling. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 12, 4 (2016), 53.

[25] Xiao Yu, Xiang Ren, Quanquan Gu, Yizhou Sun, and Jiawei Han. 2013. Collaborative filtering with entity similarity regularization in heterogeneous information networks. *IJCAI HINA* (2013).

[26] Reza Zafarani and Huan Liu. 2009. Connecting Corresponding Identities across Communities. *ICWSM* 9 (2009), 354–357.

[27] Reza Zafarani and Huan Liu. 2013. Connecting users across social media sites: a behavioral-modeling approach. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 41–49.

[28] Jiawei Zhang, Jianhui Chen, Junxing Zhu, Yi Chang, and Philip S. Yu. 2017. Link Prediction with Cardinality Constraint. In *Proceedings of the 10th ACM International Conference on Web Search and Data Mining (WSDM '17)*.

[29] Jiawei Zhang, Xiangnan Kong, and S Yu Philip. 2013. Predicting social links for new users across aligned heterogeneous social networks. In *2013 IEEE 13th International Conference on Data Mining*. IEEE, 1289–1294.

[30] Jiawei Zhang and S Yu Philip. 2015. Integrated anchor and social link predictions across social networks. In *Proceedings of the 24th International Conference on Artificial Intelligence*. AAAI Press, 2125–2131.

[31] Jiawei Zhang and Philip S. Yu. 2015. Multiple Anonymized Social Networks Alignment. In *Proceedings of the IEEE International Conference on Data Mining*.

[32] Jiawei Zhang, Philip S. Yu, and Yuanhua Lv. 2015. Organizational Chart Inference. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[33] Jiawei Zhang, Philip S. Yu, and Yuanhua Lv. 2017. Enterprise Employee Training via Project Team Formation. In *Proceedings of the 10th ACM International Conference on Web Search and Data Mining*.

[34] Jiawei Zhang, Philip S. Yu, and Zhi-Hua Zhou. 2014. Meta-path Based Multi-network Collective Link Prediction. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1286–1295.

[35] Yutao Zhang, Jie Tang, Zhilin Yang, Jian Pei, and Philip S Yu. 2015. Cosnet: connecting heterogeneous social networks with local and global consistency. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1485–1494.