IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received June 26, 2017, accepted July 8, 2017, date of publication July 12, 2017, date of current version August 22, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2726339

CHRS: Cold Start Recommendation Across Multiple Heterogeneous Information Networks

JUNXING ZHU[1], JIAWEI ZHANG[2], CHENWEI ZHANG[3], QUANYUAN WU[1], YAN JIA[1], BIN ZHOU[1], AND PHILIP S. YU[3], (Fellow, IEEE)

[1]College of Computer, National University of Defense Technology, Changsha 410073, China
[2]Florida State University, Tallahassee, FL 32306, USA
[3]University of Illinois at Chicago, Chicago, IL 60607, USA

Corresponding author: Junxing Zhu (daishanshen5036@sina.com)

This work was supported in part by the National Key Fundamental Research and Development Program of China under Grant 2013CB329601, Grant 2013CB329604, and Grant 2013CB329606, in part by the National Key Research and Development Program of China under Grant 2016QY03D0601 and Grant 2016QY03D0603, in part by the National Natural Science Foundation of China under Grant 61502517, Grant 61372191, and Grant 61572492, and in part by the Major Pre-Research Project of the National University of Defense Technology.

ABSTRACT Nowadays, people are overwhelmingly exposed to various kinds of information from different information networks. In order to recommend users with the information entities that match their interests, many recommendation methods have been proposed so far. And some of these methods have explored different ways to utilize different kinds of auxiliary information to deal with the information sparsity problem of user feedbacks. However, as a special kind of information sparsity problem, the "cold start" problem is still a big challenge not well-solved yet in the recommendation problem. In order to tackle the "cold start" challenge, in this paper, we propose a novel recommendation model, which integrates the auxiliary information in multiple heterogeneous information networks (HINs), namely the *Cross-HIN Recommendation System (CHRS)*. By utilizing the rich heterogeneous information from meta-paths, the *CHRS* is able to calculate the similarities of information entities and apply the calculated similarity scores in the recommendation process. For the information entities shared among multiple information networks, *CHRS* transfers item latent information from other networks to help the recommendation task in a given network. During the information transfer process, *CHRS* applies a domain adaptation matrix to tackle the domain difference problem. We conduct experiments to compare our *CHRS* method with several widely employed or the state-of-art recommendation models, and the experimental results demonstrate that our method outperforms the baseline methods in addressing the "cold start" recommendation problem.

INDEX TERMS Recommendation system, "cold start" problem, heterogeneous information networks, information transfer.

I. INTRODUCTION

Nowadays people are engaged in different kinds of online information networks, e.g., academic bibliographic network [1] and online social networks [2]. And people can easily be overwhelmed by a vast amount of information entities, like movies, books and music, in these networks [3]. In order to recommend users with the information entities matching their interests, many recommendation methods have been proposed so far. However, traditional recommendation methods usually suffer from the information sparsity problem a lot, especially the sparsity of user feedback information [4]. For instance, 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 users' feedback information (e.g., review or user-item rating information) in the network, on which the recommendation task is executed, there also exists some other auxiliary information that can be used to help solve the information sparsity problem in the recommendation systems [4]. In this paper, we use the target network to denote the network on which the recommendation task is executed, and use the source network to denote the network without any recommendation task executed on it but is used as the auxiliary information data source for the

[1]www.imdb.com

recommendation task of the target network. According to the source of the auxiliary information, we can categorize the auxiliary information into two types:

- The auxiliary information in the target network besides the users' feedback information.
- The auxiliary information from the source networks that can be transferred to the target network.

Online networks are usually Heterogeneous Information Networks (HINs), which contain different types of nodes or links [5]. If the target network is a HIN, besides the users' feedback information, there could be other useful information implicitly presented in this network that could be exploited for better recommendation performance. For instance, the social connections among the users [6], the genres that the items belong to [5], and the correlation among these items [5]. Some of these information is proved to be very effective to the recommendation problem [3], [5]–[7]. For example: friends are likely to show their interests on the same items [7], and thus it is of great value to recommend items according to people's social relationships. Meanwhile, as proposed in [8], a person may show his/her favorite genre(s) of books on the Internet, and thus it makes sense to consider the genre information when recommending books to certain persons.

The auxiliary information from the external source networks can also be transferred to the target network to provide complimentary knowledge for the recommendation task. For example, assuming that the target network brings in an item from a given source network, based on the known set of users who like the item in the source network, we can recommend the item to users who share common interest with these known users in the target network as well. There are several methods can be applied to transfer information across networks for recommendation [9]–[13]. However, some of these methods are usually suffered from the ''negative transfer problem [14], which is caused by transferring information between the domains that are not related enough, and results in bad recommendation performances. Because these methods implicitly assume that the source and target domains are highly-related to each other, but in the real world, such an assumption can hardly be met. For example, when recommending topics to the Chinese history fans in Sina Weibo[2] according to the popular topics of the American history fans in Twitter, the recommendation task may not perform well. Because the American history fans are familiar with the big history events happened in the USA, while the Chinese history fans mainly focus on Chinese history events.

One way to solve the ''negative transfer'' problem is to apply anchor links to the cross-network information transfer. Anchor links are the inter-network links that connect information entities of the same users or items in different network sources [15]. Different from most of the other network links, anchor links normally follow the *one-to-one constraint* [16], 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 [17], and can be resolved with techniques like the Mx_t models that are proposed in [18]. By using these techniques, the duplicated entities in each network can be aggregated in advance to form one unique entity, and the constraint on anchor links connecting these aggregated entities will still be ''one-to-one''. Since anchor links connect entities across two different network sources, and follow the *one-to-one constraint*, the information of two different sources/domains can be directly transferred via these links. 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 [16], [19]–[23], alignment of multi-source networks [24], [25], cross-network social link prediction [26], [27], cross-network recommendation [9]–[11]. However, when transferring information across networks by the anchor links, most of the existing cross network transfer learning methods are base on the user anchor links [10], [11], [26], [27]. But in most cases, due to privacy concerns, many users' profile information is usually anoymized [24], [28]. Besides, different network sources may have different user groups. So collecting enough user anchor links between some network sources is usually infeasible, which makes these transfer learning methods hard to adapt to many application circumstances. As a result, exploring the transfer learning methods based on the anchor links that are usually sufficient and easy to acquire become important.

Many existing recommendation methods based on auxiliary information have tried to alleviate the information sparsity problem. However, the ''cold start'' problem is still a big challenge which hasn't been effectively solved by most of the existing methods. The ''cold start'' issue originates from the data sparsity problem, and it is ubiquitously observed in recommendation systems when a network newly imports some items that are associated with no user feedback. A similar problem is the ''semi-cold start'' problem, in which the newly imported items receive a few user feedbacks, but the amount of these feedbacks is too small to be useful for recommendation.

In order to better solve the ''cold-start'' and ''semi-cold start'' problem, in this paper we propose our *Cross-HIN Recommendation System (CHRS)*, which integrates the auxiliary information in both of the source and target networks for recommendation. Figure 1 shows the basic idea of *CHRS* on solving the ''cold start'' and ''semi-cold start'' issues in a movie network. From it we can see that *CHRS* firstly extracts the movie similarity information from multiple types of relations that connect the movie items in the target network, and the movie item information from the source network, and then integrates these two kinds of information in the process of movie recommendation to get better results. The main contributions of our approach are as follows:

- *Integrating auxiliary information from different sources to solve the ''cold-start'' problem:* unlike

[2] weibo.sina.com

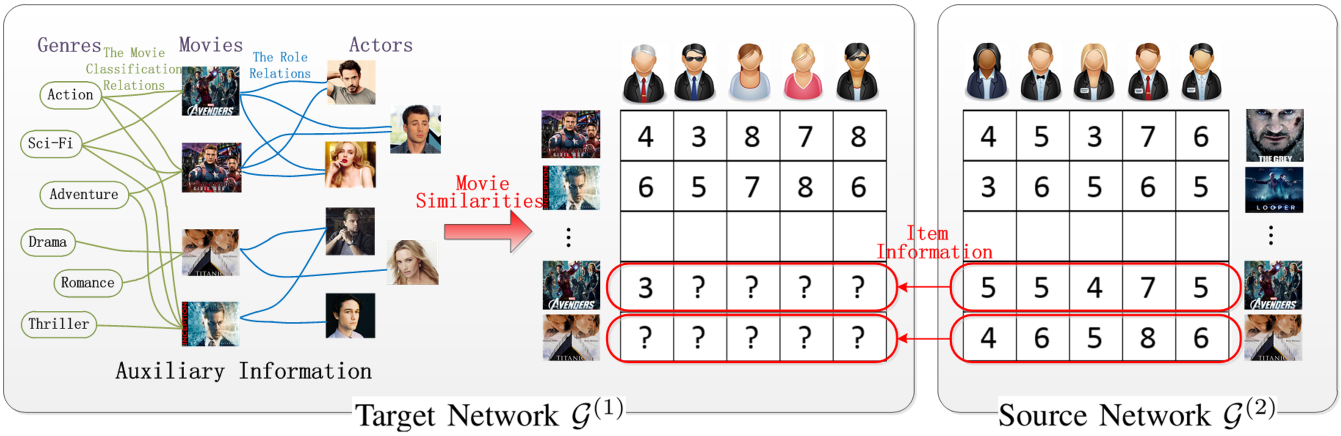


FIGURE 1. **The basic idea of *CHRS* on solving the "cold start" and "semi-cold start" issues in a movie network.**

most of the existing recommendation approaches [6], [9]–[13], [29], [30] which only focus on using the auxiliary information from only one source to solve the data sparsity problem, *CHRS* utilizes the auxiliary information not only from the target network, but also from the source network. By integrating these auxiliary information together, *CHRS* can predict the preferences of each user according to more perspectives, in this way to achieve better recommendation results and deal with the "cold-start" problem more easily.

- *Transferring information across networks via the item anchor links*: Firstly, *CHRS* adopts a proper way to transfer item information via the item anchor links, which directly connect items among different networks, and thus the "negative transfer" problem is avoided. Secondly, most of the existing transfer learning methods are based on user anchor links [10], [11], [26], [27], which make them hard to adapt to many application circumstances due to the hardship of acquiring enough user anchor links. However, *CHRS* are based on the item anchor links which can be collected more easily than user anchor links. In this way, *CHRS* can adapt to more application environments.
- *Designing a proper way to bridge the domain differences among different networks*: Although there exist a few recommendation methods which transfer information across multiple networks via the item anchor links [9], [31]. However, these methods neglect to deal with the domain differences, which are very important to the cross-network information transference in most circumstances. For example, the language used in Douban[3] is mainly Chinese, while the IMDb is dominated by English contents. Thus when transferring the movie information between these networks, we should explore a proper way to deal with the language difference. Unlike these existing transfer learning methods base on item anchor links [9], [31], *CHRS* implies a domain adaptation matrix, which can be automatically adjusted according to the extracted item information from the source and target networks during the training process, in this way to bridge the domain differences.

This paper is organized as follows. We introduce the related works in Section II. The background and preliminaries of our problem are presented in Section III. In Section IV, we introduce our *CHRS* approach. In Section V, we conduct different experiments to test the recommendation performances of *CHRS*, and analyze the experimental results. Finally, we conclude in Section VI.

II. RELATED WORKS

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 methods are widely used in many recommendation systems, and can be classified into two types of approaches: memory-based methods and model-based methods [32]. Memory-based methods make 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 [33]. Different from the menory-based collaborative filtering methods which directly use the rating values, model-based methods establish a model using the observed ratings that can interpret the given data and predict the unknown ratings [33]. Many learning models have been used for modeling the rating process, among them, matrix factorization methods, such as *Singular Value Decomposition (SVD)* [34] and the *Low Rank Matrix Factorization (LMF)* [35], are perhaps the most popular ones in recent years. And many works have explored the ways of modifying these matrix factorization models to get better performances. For example, [36] integrates a *Social Regularization* into the matrix factorization model, in this way to utilize the user similarity information to improve the recommendation performances. Reference [37] incorporates the

[3] www.douban.com

user-item subgroup analysis into the *SVD* model, to make the recommendation process distinguish the variation of users interests across different domains.

However, in some networks, user-item rating information is usually very sparse, which makes many traditional recommendation methods can't perform very well. In order to alleviate the information sparse problem, several methods have been proposed recently [3], [5]–[7], [9]–[13], [38] to apply some auxiliary information to the recommendation process. Among them, [3], [5]–[7], [39], [40] explore the way of integrating some additionally available heterogeneous information besides the users' feedback information in the target network into the matrix factorization process, in this way to have sufficient information for recommendation. For example, Yu *et al.* [5] 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 *et al.* [6] propose a flexible regularization framework, which integrates different types of the user relation information and item relation information into the recommendation process. References [39] and [40] utilize the trust relations among users to improve the recommendation performances.

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. For example, [12] and [13] try to recommend items to the users in the target network according to the preferences of users in the source network. Because anchor links can connect different networks together, via anchor links, the information closely related to the target network information can be transferred directly from the source network to the target network. But only a very few works have been done to explore the anchor link based recommendation methods. Yan *et al.* [10], [11] 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 much harder to be collected than the item anchor links, how to properly apply item anchor links to transfer information across networks in this way to improve the recommendation performances remains to be studied. Pan *et al.* [9] propose a matrix factorization framework *CST* which integrates information from different networks for recommendation via the user and item anchor links, however, the entity similarity information is not considered by their work.

Although many existing recommendation methods based on auxiliary information have tried to alleviate the information sparsity problem, however, the "cold start" problem is still a big challenge which hasn't been effectively solved by most of these methods [41]. Several works have tried to deal with the "cold start" problem in recommendation system recently [41]–[43]. For example, Li *et al.* [41] propose a trust-based recommendation model to recommend new items in newly opened shops for social network users, according to the related information of the similar shops. In this way, their method can solve the "cold start" problem. However, their method cannot be used on the network that has no shop on it, such as IMDb and Douban. Lu *et al.* [31] propose an *Amp-MF* method which transfers the information between different networks for recommendation via the anchor links, and integrates the similarity information which is computed from different auxiliary information to the regulation process to get better performances. Their experiments prove that *Amp-MF* can outperform some existing works [5], [7] on dealing with the "cold start" problem. Although the idea of *Amp-MF* looks very similar to our *CHRS* method, there are several important differences between them: 1) For *Amp-MF* the information transference between two networks is bidirectional, while for *CHRS* the information can only be transferred from the source network to the target network in the process of cross-network information transference. 2) The information used by *Amp-MF* to compute the user/item similarities in one networks includes the user-item rating information in the other network, however, *CHRS* extract the item similarities in one network only from the multiple heterogeneous information in this network. 3) *Amp-MF* never consider the domain difference problem when transferring information across networks, while *CHRS* applies proper way to deal with the domain differences. For better understanding the novelties of our *CHRS* method, we will explain the reasons of these differences between it and the *Amp-MF* when illustrating its technique details in Section III-B and Section IV, and conduct experiments to compare the performances of *CHRS* and *Amp-MF* in Section V.

Since anchor links directly connect two networks, and have great values to many cross network applications, several works on anchor link prediction have been published in the past seven years. And most of these works aim at connecting the user accounts of common users across different networks [16], [19]–[23], [44], among them: Zafarani and Liu [44] first introduce a methodology for connecting identities across social networks by usernames. Liu *et al.* [19] propose a framework to connect user accounts across heterogeneous social media platforms by using multiple user features. Kong *et al.* [16] explore the way of extracting heterogeneous features from multiple heterogeneous networks for anchor link prediction. And Zhang *et al.* [20] develop a general cross-network user alignment model which can support the integration of a number of networks.

III. BACKGROUND AND PRELIMINARIES

In this section, we illustrate the background and preliminaries of this study. However, before the illustration, we summarize the main notations used in this paper in Table 1.

A. HETEROGENEOUS INFORMATION NETWORK

A Heterogeneous Information Network (HIN) is a special type of information network, which either contains multiple types of objects or multiple types of links. Suppose $\mathcal{S} = (\mathcal{A}, \mathcal{R})$ is a network schema which consists of a set of entity

TABLE 1. Main notations.

Notation	Description
$\mathcal{S}$	A network schema
$\mathcal{A}$	A set of entity types in $\mathcal{S}$
$\mathcal{R}$	A set of relations types in $\mathcal{S}$
$\mathcal{G}$	An information network
$\mathcal{V}$	A set of entities in $\mathcal{G}$
$\mathcal{E}$	A set of relations in $\mathcal{G}$
φ_1	The entity type mapping function
φ_2	The link type mapping function
A_i	The ith entity type in $\mathcal{A}$
$\mathcal{R}_i$	the ith link type in $\mathcal{R}$
$\circ$	The composition operator on relations
$\mathcal{P}$	A meta-path set
P_k	The kth meta-path in $\mathcal{P}$
$\mathcal{I}$	The set of items that are rated or to be rated by the users in $\mathcal{G}$
v_i	The ith item in $\mathcal{I}$
S_{ijk}	The similarity between v_i and v_j under a given meta path P_k
$\bar{S}_k$	The average value of all S_{ijk} under a given meta path P_k
$S_k^{'} = \{S_{ijk}^{'}\}$	The normalized similarity matrix of items in $\mathcal{I}$ on path P_k
$S = \{S_{ij}\}$	The calculated similarity matrix of items in $\mathcal{I}$ on all the paths in $\mathcal{P}$
$\mathcal{G}^{(1)}$	The source network
$\mathcal{G}^{(2)}$	The target network
$\mathcal{V}^{(i)}$	The set of entities in $\mathcal{G}^{(i)}$
$\mathcal{E}^{(i)}$	The set of relations in $\mathcal{G}^{(i)}$
$\mathcal{U}^{(i)}$	The set of user entities in $\mathcal{V}^{(i)}$
$\mathcal{I}^{(i)}$	The set of items that are rated or to be rated by the users in $\mathcal{G}^{(i)}$
$U^{(i)}$	The latent semantic distribution matrix of $\mathcal{U}^{(i)}$
$V^{(i)}$	The latent semantic distribution matrix of $\mathcal{I}^{(i)}$
$R^{(i)}$	The user-item rating matrix of $\mathcal{G}^{(i)}$
ϕ_1	The user-item rating mapping function
L	The set of anchor links
ϕ_2	The anchor link mapping function
$S^{(i)}$	The extracted item similarity matrix of $\mathcal{G}^{(i)}$
α, λ, γ	The regularization parameters
H	The item latent domain adaptation matrix from $V^{(1)}$ to $V^{(2)}$
$\otimes$	The *Kronecker product*
T_R	The training set
T_E	The test set

types $\mathcal{A} = \{A_i\}$ and a set of relations $\mathcal{R} = \{\mathcal{R}_j\}$, where A_i and $\mathcal{R}_j$ are the ith type of entities in $\mathcal{A}$ and the jth type of relations in $\mathcal{R}$ respectively. Thus an information network is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with an entity type mapping function $\varphi_1 : \mathcal{V} \to \mathcal{A}$ and a link type mapping function: $\varphi_2 : \mathcal{E} \to \mathcal{R}$, where $\mathcal{V}$ is the set of all entities in this network, and $\mathcal{E}$ is the set of all relations in this network. If the number of entity types $|\mathcal{A}| > 1$ or the number of relation types $|\mathcal{R}| > 1$, the network is called heterogeneous information network; otherwise, it is a homogeneous information network.

Figure 2 shows the schema of a typical heterogeneous movie network. This heterogeneous network contains objects from multiple types of entities: user (U), movie (M), genre (G), writer (W), actor (A), director (D), and tag (T). And it also contains multiple types of relations. For example, in Figure 2, the link exists between user and movie denoting the user-item rating relation, between movie and actor denoting the role relation, between movie and genre denoting the movie classification relation.

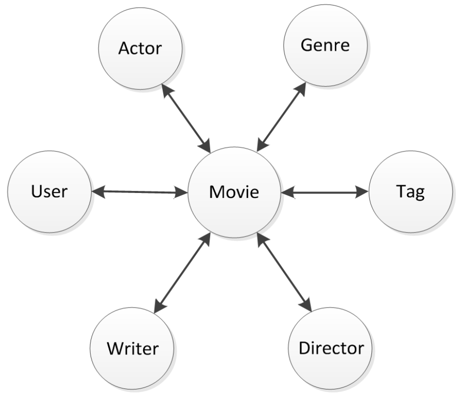


FIGURE 2. The schema of a typical heterogeneous movie network.

B. META-PATH-BASED ITEM SIMILARITY

Two entity types in a network schema of HIN can be connected via different paths, which can be called meta-path [45]. A meta path P is a path defined on the graph of a network schema $\mathcal{S} = (\mathcal{A}, \mathcal{R})$. It is denoted in the form of $A_1 \xrightarrow{\mathcal{R}_1} A_2 \xrightarrow{\mathcal{R}_2} \cdots \xrightarrow{\mathcal{R}_l} A_{l+1}$ (abbreviated as $A_1A_2 \cdots A_{l+1}$), which defines a composite relation $\mathcal{R} = \mathcal{R}_1 \circ \mathcal{R}_2 \circ \cdots \circ \mathcal{R}_l$ between type A_1 and A_{l+1}, where $\circ$ denotes the composition operator on relations. As an example shown in Figure 1, movies can be connected via "Movie-Actor-Movie" (MAM) path, "Movie-Genre-Movie" (MGM) path, "Movie-Director-Movie" (MDM) and so on. It is obvious that semantics underneath these paths are different. The MAM path means movies having common actor(s), the MGM path means movies that are in the same genre, while the MDM path means movies directed by the same director.

Several meta-path based similarity measures to quantitatively evaluate the similarities of entities in HIN have been prosed so far [30], [31], [45], [46]. Among them, the recommendation method *Amp-MF* [31] applies a method named *AmpSim* to utilize user-item rating information from different networks to compute the similarities. However, we notice that users in different domains may have different tastes [37], and any two items that are similar for the users in one network may be totally different for the users in the other network. So when the heterogeneous information is sufficient for us to extract good enough similarity information, it is not proper for us to use *AmpSim* to compute the item similarities. And since *HeteSim* [30] aims at effectively evaluating the relatedness of entities in *HIN*, and is proved to outperform many conventional measures in tasks like query, clustering and recommendation [6], [30]. In this paper, we employ *HeteSim* to evaluate the similarity of movie items. Specifically, suppose $\mathcal{I} = \{v_1, v_2, ...\}$ is the set of items that are rated or to be rated by the users in $\mathcal{G}$, S_{ijk} is the similarity between two movie items v_i and v_j under the given meta path P_k. Thus the movie similarity matrix S is determined by the given meta-path set $\mathcal{P} = \{P_1, P_2, \cdots, P_K\}$, which contains K different types of meta-path. Since the similarities computed from different types of meta-paths are different and are incomparable to each other, a Sigmoid function is used to normalize them as follows [6]:

$$S'_{ijk} = \frac{1}{1 + e^{-\beta \times (S_{ijk} - \bar{S}_k)}} \tag{1}$$

where $\bar{S}_k$ denotes the average value of all S_{ijk} under the given meta path P_k and β is set to 1. By using this function, we can confine all the movie similarity values into [0, 1] without changing their rankings, at the same time reduce the similarity difference of different paths.

Since two given movies can have different similarity values under different meta-paths, we calculate their similarity by assigning weights on the similarity values computed from different paths, and add them together. As follows:

$$S' = \sum_{k=1}^{K} \omega_k S'_k \quad \sum_{k=1}^{K} \omega_k = 1, \quad \forall \omega_k \in [0, 1] \tag{2}$$

where K is the number of selected meta-path types, $S'_k = \{S'_{ijk}\}$ denotes the normalized similarity matrix of movie items on path P_k. And ω_k represents the weight of meta-path P_k, the larger ω_k is, the more important P_k is.

For all $v_i \in \mathcal{I}$, $v_j \in \mathcal{I}$ and $P_k \in \mathcal{P}$, according to [30], we know that $S_{ijk} \leq 1.0$ and $S_{iik} = S_{jjk} = 1.0$. Thus we can infer that $S'_{ijk} < 1.0$, and $S'_{iik} = S'_{jjk} < 1.0$ according to Equation (1). Then we can further figure out that $S'_{ij} < 1.0$ and $S'_{ii} = S'_{jj} < 1.0$ according to Equation (2). However, in real world, for all $v_i \in \mathcal{I}$ the similarity of item v_i to itself should be 1.0 (i.e., $\forall i S_{ii} = 1.0$). But since $S'_{ii} < 1.0$, it is not reasonable for us to directly use S' to be the item similarity matrix S as what [6] did. And from Equation (1) and (2), we know that S' denotes the normalized similarity matrix of movie items on all paths, then for each S'_{ij} we can regard it as the normalized form of S_{ij}. So according to Equation (1), we formulize the relation between S'_{ij} and S_{ij} as follows:

$$S'_{ij} = \frac{1}{1 + e^{-\beta \times (S_{ij} - \gamma)}} \tag{3}$$

where β is set to 1, and γ is a constant which is similar to $\bar{S}_k$ in Equation (1). Since we already know $S_{11} = 1.0$ and the value of S'_{11}, we can infer that $\gamma = \ln(1 - S'_{11}) - \ln S'_{11} + S_{11}$. Then we can calculate our item similarity matrix $S = \{S_{ij}\}$ as follows:

$$S_{ij} = \gamma - \ln(1 - S'_{ij}) + \ln S'_{ij} \tag{4}$$

C. RECOMMENDATION CROSS HINs

Suppose there are two different HINs, where $\mathcal{G}^{(1)} = (\mathcal{V}^{(1)}, \mathcal{E}^{(1)})$ represents the source network and $\mathcal{G}^{(2)} = (\mathcal{V}^{(2)}, \mathcal{E}^{(2)})$ represents the target network. Let $\mathcal{V}^{(i)}$ and $\mathcal{E}^{(i)}$ denote the set of entities in $\mathcal{G}^{(i)}$ and the set of relations in $\mathcal{G}^{(i)}$ respectively, $i \in \{1, 2\}$. Suppose $\mathcal{U}^{(1)} = \{u_1^{(1)}, u_2^{(1)}, \ldots, u_a^{(1)}\}$ is the set of users in $\mathcal{G}^{(1)}$, and $\mathcal{I}^{(1)} = \{v_1^{(1)}, v_2^{(1)}, \ldots, v_b^{(1)}\}$ is the set of items that are rated or to be rated by the users (in the following part of this paper, we directly use items to represent this kind of items) in $\mathcal{G}^{(1)}$, we have $\mathcal{U}^{(1)} \cup \mathcal{I}^{(1)} \subseteq \mathcal{V}^{(1)}$. Here, $|\mathcal{U}^{(1)}| = a$ and $|\mathcal{I}^{(1)}| = b$. Let $\mathcal{E}_r^{(1)} \subset \mathcal{E}^{(1)}$ denote the set of ratings between users and items in $\mathcal{G}^{(1)}$, we have $\mathcal{E}_r^{(1)} \subseteq \mathcal{U}^{(1)} \times \mathcal{I}^{(1)}$. $\mathcal{G}^{(2)}$ is defined in a similar way, where $|\mathcal{U}^{(2)}| = c$, $|\mathcal{I}^{(2)}| = d$ and $\mathcal{E}_r^{(2)} \subseteq \mathcal{U}^{(2)} \times \mathcal{I}^{(2)}$.

Since each user may assign a group of rating values to a group of items, there exists an user-item rating mapping function ϕ_1, which maps $\mathcal{E}_r^{(1)}$ and $\mathcal{E}_r^{(2)}$ to the user-item rating matrices $R^{(1)}$ and $R^{(2)}$, where $R^{(1)} = [R_{i,j}^{(1)}]_{a \times b}$ is an $a \times b$ matrix that refers to a users' ratings on b items in $\mathcal{G}^{(1)}$. Similarly, $R^{(2)} = [R_{i,j}^{(2)}]_{c \times d}$ is an $c \times d$ matrix that refers c users' ratings on d items 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 $L \subseteq \mathcal{I}^{(1)} \times \mathcal{I}^{(2)}$ is the set of anchor links, which connect the items in $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$. Thus there must exist an anchor link mapping function $\phi_2 : L \to T$, where matrix $T = [T_{i,j}]_{b \times d}$. Supposing that $l(v_i^{(1)}, v_j^{(2)}) \in L$ is an anchor link that connects the ith item in $\mathcal{I}^{(1)}$ with the jth item in $\mathcal{I}^{(2)}$. Thus if $l(v_i^{(1)}, v_j^{(2)})$ exists, $T_{i,j}$ is set to 1; otherwise, $T_{i,j}$ is set to 0. Since all of the anchor links in $\mathcal{A}$ follow the one-to-one constraint, to all of the $T_{i,j} \in T$, we have: $\forall i, \forall j \left(\sum_{k=0}^{d} T_{i,k} \leq 1, \sum_{k=0}^{b} T_{k,j} \leq 1\right)$.

The goal of *Recommendation Cross HINs* is to utilize different types of information from different HINs to improve the effects of predicting the missing rating values in the target network user-rating matrix $R^{(2)}$. The utilized information includes: 1)the heterogeneous information that can be used to evaluate the similarities between all the items 2) the item information transferred from the source network user-rating matrix $R^{(2)}$ via T.

IV. CHRS: OUR TWO-STEP SOLUTION FOR RECOMMENDATION CROSS HINs

As we discussed in Section II, the recently proposed *Amp-MF* method uses a bidirectional way to transfer information between two networks. However, the bidirectional information transference may not adapt to the "cold start" problem discussed in this paper. For example, if the target network $\mathcal{G}^{(2)}$ newly imports some items which are very popular and have sufficient user feedbacks in the source network $\mathcal{G}^{(1)}$, there is very little or no user feedback information of these items in $\mathcal{G}^{(2)}$. So from the user feedback information in $\mathcal{G}^{(2)}$, it is hard for us to extract the information which can precisely represent the latent semantic distribution of these newly imported items (i.e., these extracted latent semantic distribution information has poor quality). However, from the sufficient user feedbacks in $\mathcal{G}^{(1)}$, we can extract the information that can precisely represent these items' latent semantic distribution (i.e., these extracted latent semantic distribution information has good quality). let $V^{(1)}$ and $V^{(2)}$ denote the extracted latent semantic distribution information of these items in $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$ respectively. Through the bidirectional information transference of *Amp-MF*, which

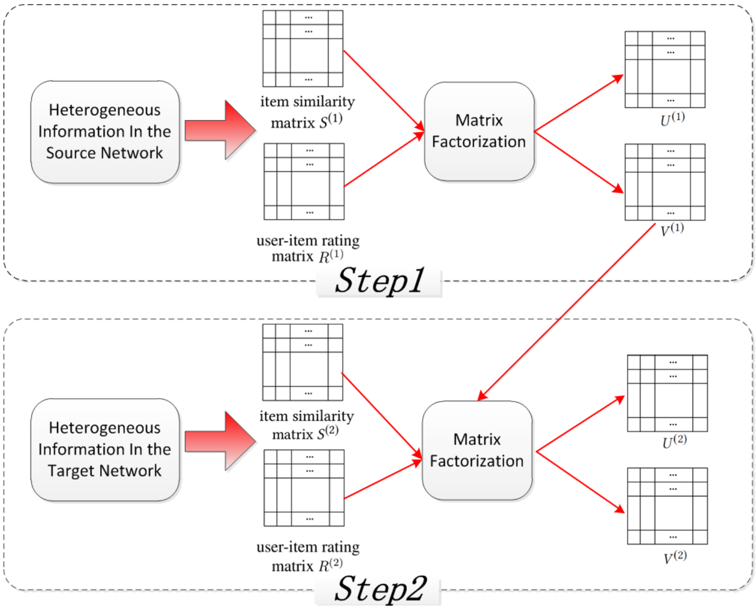


FIGURE 3. **The main framework of our *CHRS* approach.**

makes $V^{(1)}$ and $V^{(2)}$ be as similar as possible when extracting them at the same time, the good quality of $V^{(1)}$ will decline according to $V^{(2)}$. Then the improvement of $V^{(2)}$'s quality will be limited by the declined quality of $V^{(1)}$. So in order to avoid the quality declining of the information which is extracted from the source network and then transfered to the target network, we propose our *Cross-HIN Recommendation System (CHRS)*. It applies a two steps process to ensure only the information in the source network can be transferred to the target network as follows:

- *Step 1*: utilize matrix factorization framework to incorporate item similarity information in the source network $\mathcal{G}^{(1)}$, in this way to extract the latent factors of the items in $\mathcal{G}^{(1)}$.
- *Step 2*: Transfer the extracted item latent factors from $\mathcal{G}^{(1)}$ to the target network $\mathcal{G}^{(2)}$ via the item anchor links, and integrate them with the item similarity information and user-item rating information to conduct recommendation for the users in $\mathcal{G}^{(2)}$.

The main framework of *CHRS* is illustrated in Figure 3, and the details will be shown in the following subsections.

A. STEP 1: ITEM LATENT FACTOR EXTRACTION IN THE SOURCE NETWORK

The *Low-Rank Matrix Factorization* [35] has been widely studied in recommendation systems. The basic idea of it in recommendation system is to factorize the user-item rating matrix R into two matrices (U and V), representing user 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:

$$\min_{U,V} L = \frac{1}{2}\sum_{i,j} W_{i,j}\left(R_{i,j} - U_i V_j^T\right)^2 + \frac{\lambda}{2}\left(\|U\|^2 + \|V\|^2\right) \tag{5}$$

where $W = [W_{i,j}]$ is a corresponding nonnegative weight matrix, if user i has rated item j, then $W_{i,j} = 1$; otherwise, $W_{i,j} = 0$. And to a given matrix $\mathbb{R}$ ($\mathbb{R}$ can be U, V, W, or R, etc.), $\mathbb{R}_k$ 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.

Since similar items have similar features and are easy to get similar ratings by the same user. That is, two items v_i and v_j with high similarity value $S_{i,j}$ are more likely to have similar latent factor representations V_i and V_j. Based on this assumption, several recommendation approaches have explored the ways to integrate *Item Similarity Regularization* to the matrix factorization [5], [6]. According to the theories in [5] and [6], we propose the objective function $\mathcal{L}^{(1)}$ for the source network $\mathcal{G}^{(1)}$ as follows:

$$\begin{aligned}\min_{U^{(1)},V^{(1)}} \mathcal{L}^{(1)} &= \frac{1}{2}\sum_{i=0}^{a}\sum_{j=0}^{b} W_{i,j}^{(1)}\left(R_{i,j}^{(1)} - U_i^{(1)} V_j^{(1)T}\right)^2 \\ &\quad + \frac{\alpha}{2}\sum_{i=0}^{a}\sum_{j=0}^{b} S_{i,j}^{(1)}\|V_i^{(1)} - V_j^{(1)}\|^2 \\ &\quad + \frac{\lambda}{2}\left(\|U^{(1)}\|^2 + \|V^{(1)}\|^2\right)\end{aligned} \tag{6}$$

In this function, 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 $V^{(1)} \in \mathbb{R}^{b\times m}$, where m is the dimension of latent factors in $U^{(1)}$ and $V^{(1)}$, and $m \ll \min(a,b)$. $W^{(1)} = [W^{(1)}_{i,j}]$ is the corresponding nonnegative weight matrix of $\mathcal{G}^{(1)}$, and $S^{(1)} \in \mathbb{R}^{b\times b}$ is the extracted item similarity matrix of $\mathcal{G}^{(1)}$. $\frac{\alpha}{2}\sum_{i=0}^{a}\sum_{j=0}^{b} S^{(1)}_{i,j}\|V^{(1)}_i - V^{(1)}_j\|^2$ is the *Item Similarity Regularization* term, which makes sure that for any two items v_i and v_j with high similarity value $S_{i,j}$, the difference between their latent factor representations V_i and V_j should be small enough. And similar to Equation (5), λ represents the regularization parameter that is used to adjust the importance of the quadratic regularization term. α represents the regularization parameter that is used to adjust the importance of *Item Similarity Regularization*, and if we think that the *Item Similarity Regularization* process is more important than the matrix factorization process, we can assign α with a value which is bigger than 1.0.

In order to simplify the optimization process, we rewrite the *Item Similarity Regularization* term in Equation (6) (i.e., α term) into trace form. Suppose $M^{(1)} = D^{(1)} - S^{(1)}$ and $D^{(1)}$ is a diagonal matrix with elements $D^{(1)}_{i,i} = \sum_j S^{(1)}_{i,j}$. The trace form of the *Item Similarity Regularization* term in Equation

$$\begin{aligned}
&\frac{\alpha}{2}\sum_{i=0}^{a}\sum_{j=0}^{b} S^{(1)}_{i,j}\|V^{(1)}_i - V^{(1)}_j\|^2 \\
&\quad = \alpha\Big\{\sum_{i=0}^{a}\sum_{j=0}^{b} V^{(1)}_i S^{(1)}_{i,j} V^{(1)T}_i - \sum_{i=0}^{a}\sum_{j=0}^{b} V^{(1)}_i S^{(1)}_{i,j} V^{(1)T}_j\Big\} \\
&\quad = \alpha\Big\{\sum_{i=0}^{a} V^{(1)}_i D^{(1)}_{i,i} V^{(1)T}_i - \sum_{i=0}^{a}\sum_{j=0}^{b} V^{(1)}_i S^{(1)}_{i,j} V^{(1)T}_j\Big\} \\
&\quad = \alpha Tr\big(V^{(1)T} M^{(1)} V^{(1)}\big)
\end{aligned} \tag{7}$$

As we discussed before, $S^{(1)}$ can be calculated from the heterogeneous information of $\mathcal{G}^{(1)}$ by Equation (2). To solve this equation, [5] uses a supervised weight learning method to automatically determine the weights of meta paths. Different from [5], the approach in [6] firstly uses Equation (1) to normalize the similarity values evaluated from each meta-path, and then sets all the meta-path weights with the equal value. In this way, [6] can calculate good enough item similarities for the recommendation systems and at the same time save the cost of training the meta-path weights. So we consider the way in [6], and use Equations 1-4 to compute the meta-path based similarity matrix $S^{(1)}$. However, we notice that [6] considers the item similarities not only from the relations without any values on them (e.g., the relations between movies and actors, the relations among users), but also the relations which have their own values (e.g., the user-item rating relations). Thus it can result in unreasonable similarity values. For example, *MUM* is an important meta-path used by [6] to compute item similarities, and a *MUM* path is formed by two user-item rating relations (each of which has its own rating value) and represents two movie items rated by the same user(s). Suppose there are two movie item v_i and v_j, both of which are rated by only one user u_k, where the rating values assigned to v_i and v_j by u_k are 1 and 9 respectively. Thus we can infer that v_i and v_j are likely to be different types of movies, since the same user u_k shows totally different preferences on them. But by using the method in [6], the computed similarity value between v_i and v_j through meta-path *MUM* is 1, which means v_i and v_j are almost the same to u_k, that is in contradiction to the fact. So in our approach, the relations which have their own values will not be considered when computing the item similarities (i.e., meta-paths like *MUM* will not be considered).

Since the objective function in Equation (6) is non-convex, we adopt an iterative optimization algorithm that alternatively optimizes each variable while fixing others until convergence [47]. Specifically, by calculating the partial derivatives of the objective $\mathcal{L}^{(1)}$ with respect to $U^{(1)}$ and $V^{(1)}$ respectively, and setting them to 0, we have:

$$\begin{cases}
\dfrac{\partial\mathcal{L}}{\partial U^{(1)}} = \left(W^{(1)} * \left(U^{(1)}V^{(1)T} - R^{(1)}\right)\right)V^{(1)} \\
\qquad + \lambda U^{(1)} = 0 \\
\dfrac{\partial\mathcal{L}}{\partial V^{(1)}} = \left(W^{(1)T} * \left(V^{(1)}U^{(1)T} - R^{(1)T}\right)\right)U^{(1)} \\
\qquad + \lambda V^{(1)} + 2\alpha M^{(1)}V^{(1)} = 0
\end{cases} \tag{8}$$

By transforming all the equations in Equation (8) into their corresponding linear system forms, we can solve them directly. For example, supposing that the *vec* operator reshapes a matrix $A = [a_1, a_2, \ldots, a_n]$ to its column vector form $vec(A) = [a_1^T, a_2^T, \ldots, a_n^T]^T$ 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 3 matrices, $\otimes$ is the *Kronecker product*. We can rewrite $\frac{\partial\mathcal{L}^{(1)}}{\partial U^{(1)}} = 0$ as a linear system:

$$AX = B \tag{9}$$

where $A = \left(V^{(1)T} \otimes I_1\right)diag\left(vec(W^{(1)})\right)(V^{(1)} \otimes I_1) + \lambda$, $X = vec(U^{(1)})$, $B = \left(V^{(1)T} \otimes I_1\right)diag\left(vec(W^{(1)})\right)vec(R^{(1)})$, 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 $V^{(1)}$ by solving its corresponding $AX = B$ forms in the same way. However, for each of $U^{(1)}$ and $V^{(1)}$, the computation of its related A^{-1} is usually time consuming. Alternatively, we can solve Equation (8) iteratively by using the conjugate gradient(CG) method [48], which only needs to perform matrix multiplications on the equations in Equation (8) 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.

Thus the whole procedure of *Step 1* is summarized in Algorithm 1. Via this algorithm, we can extract the items' latent semantic distribution matrix in $\mathcal{G}^{(1)}$ and transfer it to the target network $\mathcal{G}^{(2)}$, in this way to improve the recommendation performance in $\mathcal{G}^{(2)}$.

Algorithm 1 Algorithm Framework of *Step 1* of *CHRS*

Input: $\mathcal{G}^{(1)}$: the source heterogeneous information network; α, λ:controlling parameters defined above; $\mathcal{P}$: the set of given meta-paths used to calculate item similarities; ϕ_1: the mapping function defined above;

Output: $U^{(1)}$: the users' latent semantic distribution matrix in $\mathcal{G}^{(1)}$; $V^{(1)}$: The items' latent semantic distribution matrix in $\mathcal{G}^{(1)}$;

1: Use ϕ_1 to create the user-item rating matrices $R^{(1)}$ according to the rating information in $\mathcal{G}^{(1)}$
2: Create the weight matrix $W^{(1)}$ according to the rating information in $\mathcal{G}^{(1)}$
3: Create the item similarity matrix $S^{(1)}$ according to $\mathcal{P}$ and the related relation information in $\mathcal{G}^{(1)}$ by the method in Section III-B
4: Calculate $M^{(1)}$ by $S^{(1)}$
5: Initialize $U^{(1)}$ and $V^{(1)}$
6: **repeat**
7: Update $U^{(1)}$ by solving $\frac{\partial \mathcal{L}^{(1)}}{\partial U^{(1)}} = 0$ in Eq. (8)
8: Update $V^{(1)}$ by solving $\frac{\partial \mathcal{L}^{(1)}}{\partial V^{(1)}} = 0$ in Eq. (8)
9: **until** Eq. (6 converges)

B. STEP 2: RECOMMENDATION PROCESS IN THE TARGET NETWORK

In *Setp 2*, we transfer the extracted item latent factors from $\mathcal{G}^{(1)}$ to the target network $\mathcal{G}^{(2)}$ via the item anchor links, and integrate them with the item similarity information and user-item rating information to conduct recommendation.

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 [9]. 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. And most of the traditional cross-network recommendation methods that are based on anchor links deal with the information transference in this way. E.g., the *CST* method [9] and the *Amp-MF* method [31]. However, since there may exist some domain differences between two networks (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 our *CHRS* approach novelly applies 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. So the regularization term on the transfer of item latent factors between $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$ is as follows:

$$\frac{1}{2}\|T^T V^{(1)} H - T^T T V^{(2)}\|^2 \tag{10}$$

Similar to Equation (6), the objective function $\mathcal{L}^{(2)}$ for the recommendation task in the target network $\mathcal{G}^{(2)}$ should also integrate item similarity information to the matrix factorization. By adding Equation (10), which is the regularization term on the item latent factors transfer, to the objective function $\mathcal{L}^{(2)}$, we have:

$$\begin{aligned}\min_{U^{(2)},V^{(2)},H} \mathcal{L}^{(2)} = & \frac{1}{2}\sum_{k=0}^{c}\sum_{l=0}^{d} W_{k,l}^{(2)}\left(R_{k,l}^{(2)} - U_k^{(2)} V_l^{(2)T}\right)^2 \\ & + \alpha Tr\left(V^{(2)T} M^{(2)} V^{(2)}\right) \\ & + \frac{\beta}{2}\|T^T V^{(1)} H - T^T T V^{(2)}\|^2 \\ & + \frac{\lambda}{2}\left(\|U^{(2)}\|^2 + \|V^{(2)}\|^2 + \|H\|^2\right)\end{aligned} \tag{11}$$

where $R^{(2)} \in \mathbb{R}^{c\times d}$, $U^{(2)} \in \mathbb{R}^{c\times n}$ and $V^{(2)} \in \mathbb{R}^{d\times n}$. n is the dimension number of latent factors in $U^{(2)}$ and $V^{(2)}$, and $n \ll \min(c, d)$. $W^{(2)} = \{W_{k,l}^{(2)}\}$ is the corresponding nonnegative weight matrix of $\mathcal{G}^{(2)}$, and $S^{(2)} \in \mathbb{R}^{d\times d}$ is the item similarity matrix of $\mathcal{G}^{(2)}$ which is computed in the same way as computing $S^{(1)}$. $M^{(2)} = D^{(2)} - S^{(2)}$ and $D^{(2)}$ is a diagonal matrix with elements $D_{i,i}^{(2)} = \sum_j S_{i,j}^{(2)}$. α and λ have similar roles to their roles in Equation (6), while β is the regulation parameter relates to the importance of the regularization term on the transfer of item latent factors. And if we think that the transfer of item latent factors is more important than the matrix factorization process, we can assign β with a value which is bigger than 1.0.

Since the objective function in Equation (11) is non-convex, we adopt an iterative optimization algorithm that alternatively optimizes each variable while fixing others until convergence as we did in *Step 1*. Specifically, by calculating the partial derivatives of the objective $\mathcal{L}^{(2)}$ with respect to $U^{(2)}$, $V^{(2)}$ and H respectively, and setting them to 0, we have:

$$\begin{cases}\frac{\partial \mathcal{L}^{(2)}}{\partial U^{(2)}} = \left(W^{(2)} * \left(U^{(2)} V^{(2)T} - R^{(2)}\right)\right) V^{(2)} \\ \qquad + \lambda U^{(2)} = 0 \\ \frac{\partial \mathcal{L}^{(2)}}{\partial V^{(2)}} = \left(W^{(2)T} * \left(V^{(2)} U^{(2)T} - R^{(1)T}\right)\right) U^{(2)} \\ +2\alpha M^{(2)} V^{(2)} - \beta\left(T^T V^{(1)} H - T^T T V^{(2)}\right) \\ \qquad + \lambda V^{(2)} = 0 \\ \frac{\partial \mathcal{L}^{(2)}}{\partial H} = \beta V^{(1)T} T\left(T^T V^{(1)} H - T^T T V^{(2)}\right) \\ \qquad + \lambda H = 0\end{cases} \tag{12}$$

Similar to solve Equation (8), we can solve the equations in Equation (12) iteratively by the conjugate gradient method, in this way to update $U^{(2)}$, $V^{(2)}$ and H.

Thus the whole procedure of *Step 2* is summarized in Algorithm 2. Via this algorithm, the domain difference between the $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$ is bridged by the domain adaptation matrix H, and item similarities in $\mathcal{G}^{(2)}$ are calculated from the heterogeneous information in $\mathcal{G}^{(2)}$ by the method in Section III-B. Then the extracted items' latent semantic distribution matrix in $\mathcal{G}^{(1)}$ and item similarity matrix in $\mathcal{G}^{(2)}$ are properly applied to help computing the items' latent semantic distribution matrix in $\mathcal{G}^{(2)}$. And since the information transference between $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$ follows the two step rule: firstly, extract information from $\mathcal{G}^{(1)}$, then use the extracted information to help the recommendation task in $\mathcal{G}^{(2)}$; the information extracted from the limited user feedbacks in the target network $\mathcal{G}^{(2)}$ cannot affect the extracted information in the source network.

Algorithm 2 Algorithm Framework of *Step 2* of *CHRS*

Input: $V^{(1)}$: the computed item latent semantic distribution matrix in the *setp 1* of *CHRS*; $\mathcal{G}^{(2)}$: the source heterogeneous information network; α, β, λ:controlling parameters defined above; $\mathcal{P}$: the set of given meta-paths used to calculate item similarities; L: the set of given anchor links; ϕ_1,ϕ_2: mapping functions defined above;

Output: $U^{(2)}$: the users' latent semantic distribution matrix in $\mathcal{G}^{(2)}$; $V^{(2)}$: The items' latent semantic distribution matrix in $\mathcal{G}^{(2)}$;

Use ϕ_1 to create the user-item rating matrices $R^{(2)}$ according to the rating information in $\mathcal{G}^{(2)}$

Use ϕ_2 to create matrix T according to L

Create the weight matrix $W^{(2)}$ according to the rating information in $\mathcal{G}^{(2)}$

Create the item similarity matrix $S^{(2)}$ according to $\mathcal{P}$ and the related relation information in $\mathcal{G}^{(2)}$ by the method in Section III-B

Calculate $M^{(2)}$ by $S^{(2)}$

Initialize $U^{(2)}$, $V^{(2)}$ and H

repeat

  Update $U^{(2)}$ by solving $\frac{\partial \mathcal{L}^{(2)}}{\partial U^{(2)}} = 0$ in Eq. (12)

  Update $V^{(2)}$ by solving $\frac{\partial \mathcal{L}^{(2)}}{\partial V^{(2)}} = 0$ in Eq. (12)

  Update H by solving $\frac{\partial \mathcal{L}^{(2)}}{\partial H} = 0$ in Eq. (12)

until Eq. (11 converges)

V. EXPERIMENT

In this section, we conduct several experiments to compare the proposed *CHRS* approach with several state-of-art recommendation methods, in this way to verify the superiority of our approach.

A. DATA PREPARATION

We crawl our experimental datasets from two HINs $\mathcal{G}^{(a)}$ and $\mathcal{G}^{(b)}$. $\mathcal{G}^{(a)}$ denotes Douban Movie.[4] 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. $\mathcal{G}^{(b)}$ represents 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 $\mathcal{I}^{(a)} \subset \mathcal{G}^{(a)}$ and $\mathcal{I}^{(b)} \subset \mathcal{G}^{(b)}$. These links are crawled by tracing the property of "IMDb Link" on the homepage of each movie in Douban Movie. So in total, we have crawled the related heterogeneous information of $14,831$ movies and $808,322$ users from Douban Movie, together with the heterogeneous information of $14,485$ movies and $557,821$ users from IMDb. We also collect $25,065$ item anchor links between movies in these two networks.

In our experiment, in order to make sure that each user and each item have enough rating information, we select our experimental data from the crawled datasets as follows: Firstly, in the crawled data set of each network, we randomly select 800 active users who have rated more than 80 films, and form the experimental user sets $\mathcal{U}^{(a)} \subset \mathcal{G}^{(a)}$ and $\mathcal{U}^{(b)} \subset \mathcal{G}^{(b)}$ respectively. Then we select 800 movies, each of which is rated by more than 25 users in $\mathcal{U}^{(a)}$ and $\mathcal{U}^{(b)}$ simultaneously, and form the experimental item entities sets $\mathcal{I}^{(a)} \subset \mathcal{G}^{(a)}$ and $\mathcal{I}^{(b)} \subset \mathcal{G}^{(b)}$ (here, $\mathcal{I}^{(a)}$ and $\mathcal{I}^{(b)}$ share the same items), as well as the corresponding movie anchor link set $\mathcal{A}$. Moreover, we collect the heterogeneous information shown in Fig. 2 for each selected movie. 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 map $\mathcal{E}^{(k)}$ to the original user rating matrix $\dot{R}^{(k)}$ by the user-item rating mapping function ϕ_1 (ϕ_1 is defined in Section III-C), where $k \in a, b$. However, since the $\dot{R}^{(b)}_{i,j} \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, ?\}$ while $\dot{R}^{(a)}_{i,j} \in \{10, 20, 30, 40, 50, ?\}$ (the question mark "?" denotes a missing or unobserved rating value), we set $R^{(b)} = \dot{R}^{(b)}$ and set $R^{(a)}_{i,j}$ as follows:

$$R^{(a)}_{i,j} = \begin{cases} \dot{R}^{(a)}_{i,j}/5 & \text{if } \dot{R}^{(a)}_{i,j} \neq ? \\ ? & \text{if } \dot{R}^{(a)}_{i,j} = ? \end{cases} \tag{13}$$

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

A more detailed comparison of our selected experimental datasets are shown in Table. 2.

[4] movie.douban.com

TABLE 2. Properties of the datasets.

	property	network dataset $\mathcal{G}^{(a)}$	$\mathcal{G}^{(b)}$
# entity	user	800	800
	movie	800	800
	actor	6,597	88,586
	genre	29	29
	tag	1,908	46,545
	director	917	1,092
	writer	1,888	3,302
# relation	user-movie rating	66,829	84,794
	movie-actor	16,101	145,558
	movie-genre	6,482	6,482
	movie-tag	17,320	336,747
	movie-director	2,218	2,487
	movie-writer	3,456	6,612
	anchor link	800	800

B. COMPARED METHODS

In order to analyze the performances of the proposed approaches, we compare our methods 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)*: One of the most famous collaborative filtering methods, which recommends each item according to the rating information of its top-k nearest items [32]. And different from the traditional ways which extract item similarities from the user feedback information, by using the *HeteSim* method [30] to compute the item similarities from the *HIN* information, this method is believed to achieve better results on the "cold-start" problem. According to the previous work [32], the *Adjusted Cosine Similarity* is chosen to compute the user and item similarities in this way to get better results.
- *Low-rank Matrix Factorization (LMF)*: The method proposed by Nathan Srebro and Tommi Jaakkola [35] and has been widely studied in many recommendation systems.
- *The SimMF-I(i) method*: A state-of-art recommendation framework proposed in [6], which is based on the matrix factorization method and combines user ratings as well as item similarities for recommendation.
- *The CST method*: This is a recommendation framework, which is based on the matrix factorization method and can transfer item latent factors across different networks. Different from our *CHRS* method, this method doesn't consider item similarities, and doesn't apply the domain adaptation matrix to deal with the domain difference [29]. Since the user relation information and user anchor links are unavailable
- *The Amp-MF method*: A state-of-art recommendation cross-network framework proposed in [31], which uses anchor links as well as some heterogeneous information to help the recommendation to achieve better performances. The main three differences between it and our method are listed in Section II. And since the user relation information and user anchor links are unavailable in our problems, *Amp-MF*'s regulations that are based on user similarities and user alignment constraint [31] will not be considered in our experiments.
- *The CHRS method*: This is our proposed *Cross-HIN Recommendation System* approach.

To make fair comparisons, for *LMF, SimMF-I(i), CST, Amp-MF* and *CHRS*, we set all the dimensions of latent factors as 20 and set all the parameters used to avoid overfitting as 1.0. 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. For *Amp-MF*, the item similarities are generated according to the way in [31]. And for *Ik-NN, SimMF-I(i), CST* and *CHRS*, we ensure that they use the same meta-path based item similarities, which are generated by *HeteSim* from five meaningful meta-paths of movie. These five meta-paths include: *MAM, MGM, MTM, MDM* and *MWM*, and the length of each meta-path is smaller than 4.

C. EVALUATION METRICS

In order to evaluate the effectiveness of these compared methods, we select two different metrics in terms of 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. Thus the metric MAE is defined as:

$$MAE = \frac{1}{N_t} \sum_{(i,j,k,R_{i,j}^{(k)}) \in T_E} |R_{i,j}^{(k)} - \hat{R}_{i,j}^{(k)}| \tag{14}$$

where $R_{i,j}^{(k)}$ is the actual rating value that user $u_i^{(k)} \in \mathcal{G}^{(k)}$ assigns to item $v_j^{(k)} \in \mathcal{G}^{(k)}$, and $\hat{R}_{i,j}^{(k)}$ denotes the predicted rating value that $u_i^{(k)}$ may assign to $v_j^{(k)}$. Particularly, $\hat{R}_{i,j}^{(k)}$ can be calculated by $U_i^{(k)} V_j^{(k)T}$ 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:

$$RMSE = \sqrt{\frac{1}{N_t} \sum_{(i,j,k,R_{i,j}^{(k)}) \in T_E} \left(R_{i,j}^{(k)} - \hat{R}_{i,j}^{(k)}\right)^2} \tag{15}$$

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

D. PERFORMANCE COMPARISONS ON THE "COLD START" PROBLEMS

In this subsection, we conduct two group of experiments to compare the performances of the experimental methods on the "cold start" problem.

In the first group of experiments, we assume the target network $\mathcal{G}^{(2)}$ newly imports a certain amount of movie items from the source network $\mathcal{G}^{(1)}$ (these imported items have no related user feedbacks in $\mathcal{G}^{(2)}$), and the sparsity of the existed

TABLE 3. Performance comparisons on different target network datasets, where r_t is the training ratio.

Target Network	r_t	Metrics	Ik-NN	Amp-MF	LMF	SimMF-I(i)	CST	CHRS
Douban Movie	0.2	MAE	1.4884	**1.4299**	7.5531	1.4990	1.7861	1.4362
		RMSE	1.8502	**1.7683**	7.7797	1.8491	2.2759	1.7801
	0.4	MAE	1.4594	1.4147	7.5531	1.4832	1.6842	**1.3911**
		RMSE	1.8114	1.7519	7.7797	1.8229	2.1601	**1.7286**
	0.6	MAE	1.4498	1.4129	7.5531	1.4818	1.5920	**1.3736**
		RMSE	1.7992	1.7520	7.7797	1.8154	2.0318	**1.7120**
	0.8	MAE	1.4432	1.4052	7.5531	1.4807	1.5293	**1.3589**
		RMSE	1.7900	1.7431	7.7797	1.8107	1.9481	**1.6954**
	1.0	MAE	1.4394	1.4076	7.5531	1.4817	1.4893	**1.3527**
		RMSE	1.7863	1.7467	7.7797	1.8093	1.8920	**1.6880**
IMDb	0.2	MAE	1.5071	1.4565	7.1202	1.5264	1.8328	**1.4355**
		RMSE	2.0003	1.9027	7.4287	1.9884	2.3983	**1.8925**
	0.4	MAE	1.4619	1.4231	7.1202	1.4897	1.6864	**1.3848**
		RMSE	1.9243	1.8571	7.4287	1.9308	2.2422	**1.8205**
	0.6	MAE	1.4434	1.4102	7.1202	1.4783	1.6188	**1.3634**
		RMSE	1.8969	1.8440	7.4287	1.9128	2.1654	**1.7966**
	0.8	MAE	1.4346	1.4089	7.1202	1.4780	1.5731	**1.3531**
		RMSE	1.8855	1.8401	7.4287	1.9095	2.1101	**1.7838**
	1.0	MAE	1.4296	1.4040	7.1202	1.4770	1.5316	**1.3500**
		RMSE	1.8786	1.8359	7.4287	1.9067	2.0559	**1.7803**

TABLE 4. Performance comparisons on different target network datasets, where r_a is the ratio of newly imported movie items.

Target Network	r_a	Metrics	Ik-NN	Amp-MF	LMF	SimMF-I(i)	CST	CHRS
Douban Movie	0.8	MAE	1.4853	1.5669	7.5661	1.5243	1.7633	**1.4235**
		RMSE	1.8406	1.9410	7.7941	1.8593	2.2486	**1.7816**
	0.6	MAE	1.4573	1.4827	7.5484	1.5006	1.6318	**1.3903**
		RMSE	1.8089	1.8605	7.7781	1.8323	2.0893	**1.7447**
	0.4	MAE	1.4454	1.4789	7.5854	1.4931	1.5408	**1.3589**
		RMSE	1.7884	1.8580	7.8110	1.8179	1.9629	**1.6912**
	0.2	MAE	1.4388	1.4662	7.5662	1.4823	1.4809	**1.3514**
		RMSE	1.7860	1.8503	7.7935	1.8104	1.8803	**1.6898**
IMDb	0.8	MAE	1.4879	1.5439	7.1165	1.5280	1.7717	**1.4196**
		RMSE	1.9690	1.9335	7.4256	1.9696	2.3213	**1.8700**
	0.6	MAE	1.4540	1.4485	7.1437	1.5026	1.6415	**1.3749**
		RMSE	1.9040	1.8587	7.4498	1.9291	2.1909	**1.8077**
	0.4	MAE	1.4296	1.4085	7.1394	1.4796	1.5851	**1.3517**
		RMSE	1.8773	1.8336	7.4447	1.9079	2.1320	**1.7826**
	0.2	MAE	1.4180	1.3967	7.1278	1.4666	1.5201	**1.3425**
		RMSE	1.8602	1.8249	7.4326	1.8896	2.0384	**1.7698**

user feedbacks in $\mathcal{G}^{(1)}$ changes in difference circumstances. Thus in order to improve recommendation performances on recommending the newly imported items in $\mathcal{G}^{(2)}$, different kinds of auxiliary information should be utilized. So in this group of experiments, we firstly partition the collected movie items in the target network $\mathcal{G}^{(2)}$ with 5 folds cross validation: one fold as $\mathcal{I}_x$ which denotes the set of the newly imported movie items, the rest 4 folds to form $\mathcal{I}_y$ which is the set of the old items that already exist in $\mathcal{G}^{(2)}$ for a time. And we use the observed ratings in $\mathcal{G}^{(2)}$ which relate to the items in $\mathcal{I}_x$ to form the test set T_E. Secondly, we set a training ratio r_t (r_t denotes the degree of information sparsity in the target network, the smaller it is, the sparser the existed user-item ratings in $\mathcal{G}^{(2)}$ are.). We then randomly sample r_t of the ratings relate to the items in $\mathcal{I}_y$ to form the set T_T. Thirdly, we combine T_T with the set of all the observed ratings in the source network $\mathcal{G}^{(1)}$ to form the training set T_R. The value of r_t is selected from {0.2, 0.4, 0.6, 0.8, 1.0}. If $\mathcal{G}^{(a)}$ is $\mathcal{G}^{(1)}$ then $\mathcal{G}^{(b)}$ is $\mathcal{G}^{(2)}$, and if $\mathcal{G}^{(a)}$ is $\mathcal{G}^{(2)}$ then $\mathcal{G}^{(b)}$ is $\mathcal{G}^{(1)}$. The results are shown in Table 3, in which the best performances on each of the evaluation criteria are listed in bold.

In the second group of experiments, we assume the original target network $\mathcal{G}^{(2)}$ is a network old enough (i.e., most of the items in $\mathcal{G}^{(2)}$ are old items, each of which has existed in $\mathcal{G}^{(2)}$ for sufficient time and has enough related user-item ratings for recommendation), and $\mathcal{G}^{(2)}$ newly imports some items from the source network $\mathcal{G}^{(1)}$. The number of these imported items varies in different experiments. And since these new items in $\mathcal{G}^{(2)}$ have no related user-item ratings, different kinds of auxiliary information should be utilized to improve the recommendation performances for these new items. So in each of these experiments, we firstly set a sample ratio r_a (r_a denotes the ratio of the newly imported movie items in all the movie items of $\mathcal{G}^{(2)}$), and randomly sample r_a of the collected movie items in $\mathcal{G}^{(2)}$. Then we use all of the collected user-item ratings in $\mathcal{G}^{(2)}$ that related to these sample items to form the test set T_E. The remaining user-item ratings in $\mathcal{G}^{(2)}$ and all the user-item ratings in $\mathcal{G}^{(1)}$ are used to form the training set T_R.

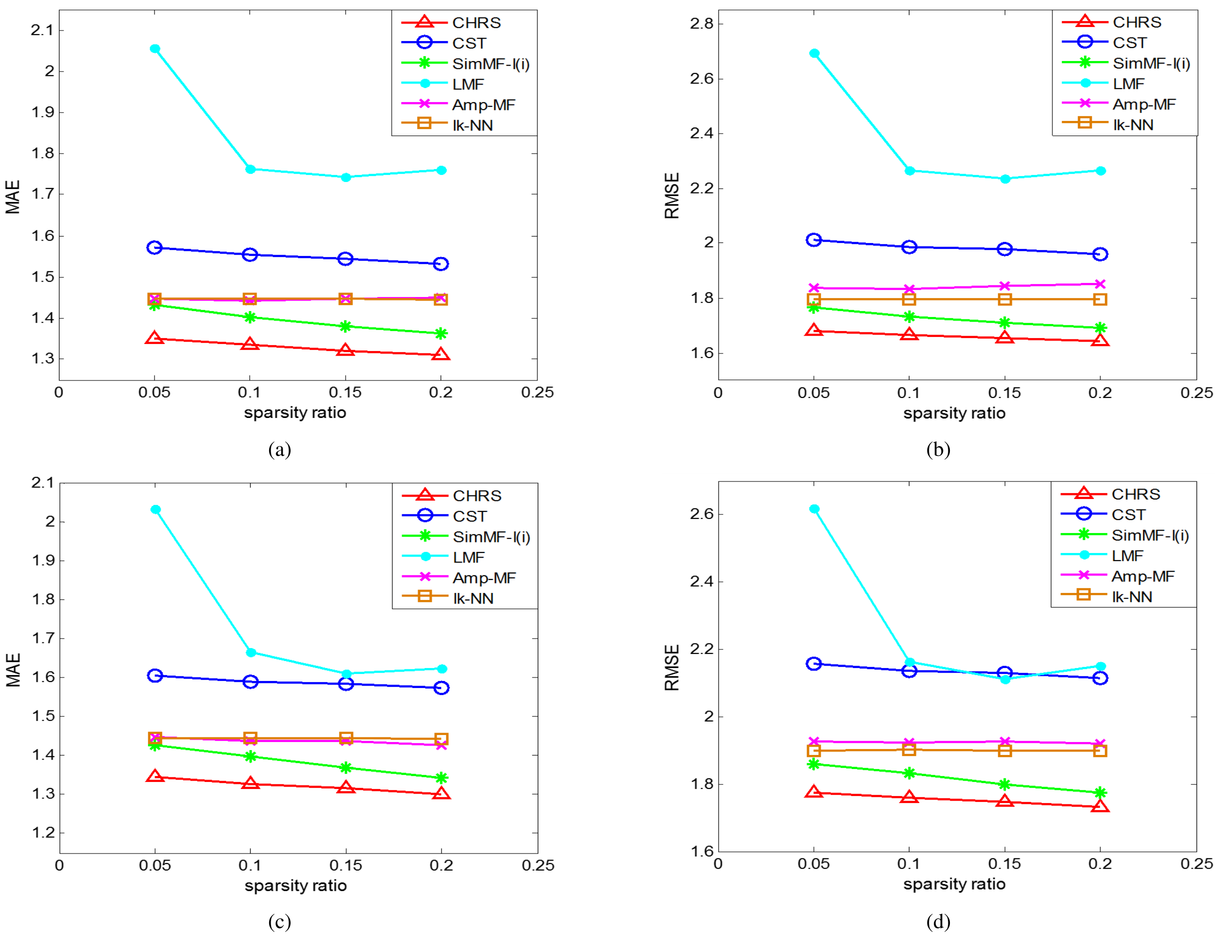


FIGURE 4. **The performance comparisons on dealing the "semi-cold start" problem in different target networks. (a) Douban Movie, MAE. (b) Douban Movie, RMSE. (c) IMDb, MAE. (d) IMDb, RMSE.**

The value of r_a is selected from $\{0.2, 0.4, 0.6, 0.8\}$. If $\mathcal{G}^{(a)}$ is $\mathcal{G}^{(1)}$ then $\mathcal{G}^{(b)}$ is $\mathcal{G}^{(2)}$, and if $\mathcal{G}^{(a)}$ is $\mathcal{G}^{(2)}$ then $\mathcal{G}^{(b)}$ is $\mathcal{G}^{(1)}$. The random sampling was carried out 5 times independently, and the average results are shown in Table 4, in which the best performances on each of the evaluation criteria are listed in bold.

According to the results in Tables 3 and 4, we can conclude that:

- The traditional *LMF* method can't solve the "cold start" problem, for the lack of important user feedback information.
- By utilizing the auxiliary information in the recommendation process, *Ik-NN*, *Amp-MF*, *SimMF-I(i)*, *CST* and *CHRS* can deal with the "cold start" problem better.
- By properly integrating different kinds of auxiliary information from different sources, our *CHRS* approach significantly outperforms the *Ik-NN*, *LMF*, *SimMF-I(i)* and *CST* methods on solving the "cold start" problem in different experimental circumstances.
- By making that only the information from the source network can be transferred to the target network, adopting a more reasonable way to compute the item similarities, and applying the domain adaptation matrix to deal with the domain difference problem, our *CHRS* can outperform *Amp-MF* in almost all the experimental "cold start" circumstances.

E. PERFORMANCE COMPARISONS ON THE "SEMI-COLD START" PROBLEM

In this subsection, we conduct experiments to investigate the performances of these experimental methods on dealing with the "semi-cold start" problem, where the target network $\mathcal{G}^{(2)}$ newly brings in some items from the source network $\mathcal{G}^{(1)}$, thus these imported items should be recommended to some users in $\mathcal{G}^{(2)}$ as soon as possible. However, the newly generated user feedback information in $\mathcal{G}^{(2)}$ that relate to these imported items may not be enough for recommendation, thus different kinds of auxiliary information should be utilized to improve the recommendation performances for these new items. So in this group of experiments, we firstly partition all the items in $\mathcal{G}^{(2)}$ with 5 folds cross validation: one fold as $\mathcal{I}_x$, which denotes the set of the newly imported items in $\mathcal{G}^{(2)}$; the other 4 folds form the set $\mathcal{I}_y$, which is the set of the old items that already exist in $\mathcal{G}^{(2)}$ and have enough related user feedback

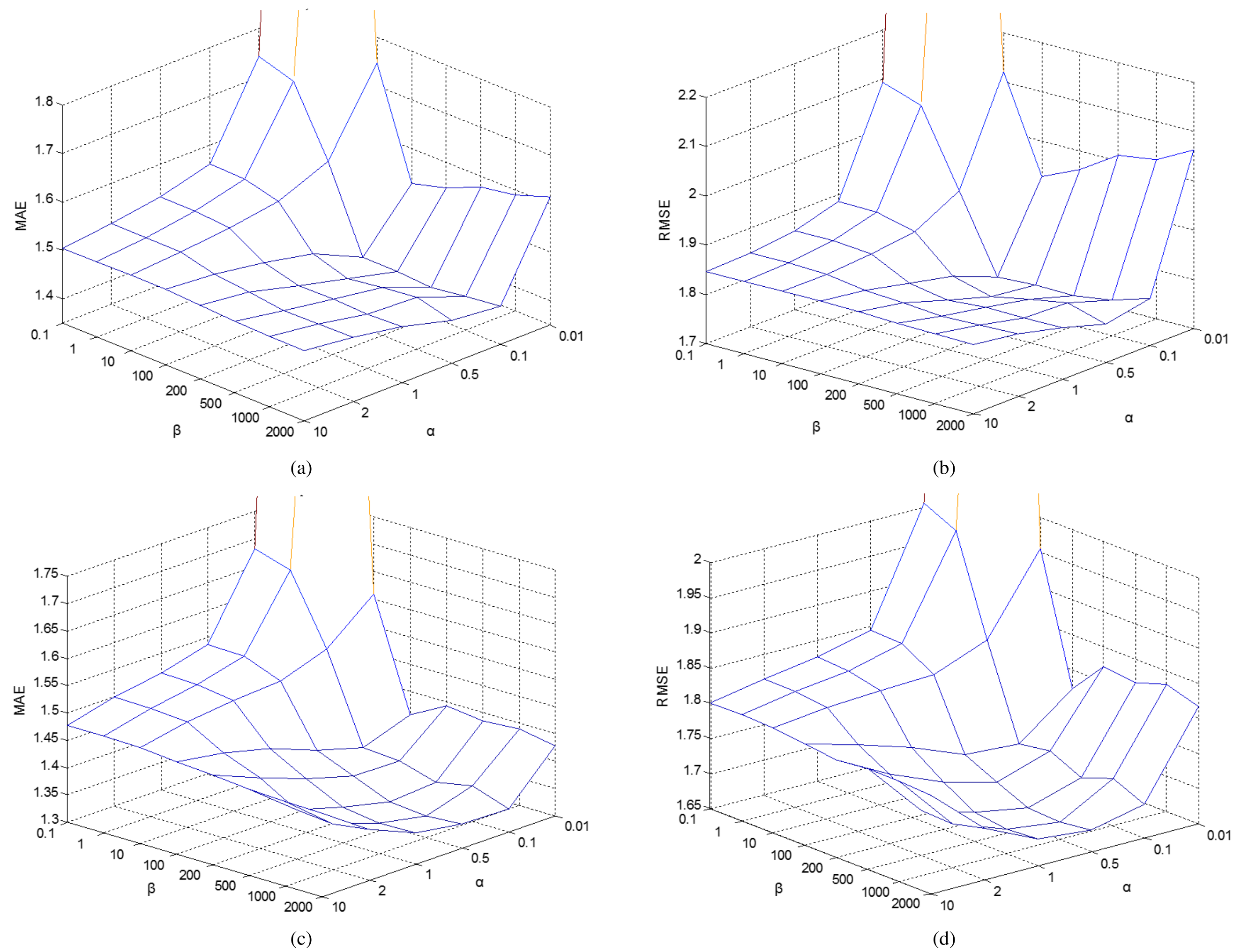


FIGURE 5. Performances of *CHRS* with varying α, β and r_t on Douban Movie dataset. The lower, the better. (a) $r_t = 0.2$, MAE. (b) $r_t = 0.2$, RMSE. (c) $r_t = 1.0$, MAE. (d) $r_t = 1.0$, RMSE.

information. Secondly, a sparsity ratio r_s is selected to denote the information sparsity degree of ratings for all the newly imported items in $\mathcal{G}^{(2)}$. The smaller r_s is, the sparser the rating information of the newly imported movies is. And to all the observed ratings in $\mathcal{G}^{(2)}$ that relate to the items in $\mathcal{I}_x$, $1 - r_s$ of them are randomly selected out to form the test set T_E, the rest of the observed ratings in $\mathcal{G}^{(2)}$ together with all the observed ratings in $\mathcal{G}^{(1)}$ are used to form the training set T_R. If $\mathcal{G}^{(a)}$ is $\mathcal{G}^{(1)}$ then $\mathcal{G}^{(b)}$ is $\mathcal{G}^{(2)}$, and if $\mathcal{G}^{(a)}$ is $\mathcal{G}^{(2)}$ then $\mathcal{G}^{(b)}$ is $\mathcal{G}^{(1)}$. To the target network, since the imported items are very new, the ratings of these items 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, in which the title of each subgraph is formed by the name of target network and the metric. From the results we can analyze:

- When the user feedback information of the recommended items is very sparse (i.e., $r_s = 0.05$), the methods integrating auxiliary information in the recommendation process can significantly outperform the traditional *LMF* methods.
- Our *CHRS* method outperforms all of the other base-line methods in different data sets. That may because *CHRS* not only applies a proper way to deal with the domain differences when transferring the information from the source network to the target network, but also integrates many important information in the recommendation process.

F. PARAMETER STUDY ON α AND β

In this subsection, we conduct parameter study for our *CHRS* on α and β, which relate to the importances of the two kinds of utilized auxiliary information respectively. On one hand, if the user-item matrices for recommendation are factorized with a very small value of α and β, *CHRS* will ignore the item similarities (i.e., the auxiliary information in the target network) and item information transfer (i.e., the auxiliary information in the source network). On the other hand, if α and β 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 α and β to achieve good performances. As a result, we will analyze how the changes of α and β effect the final recommendation accuracy in this section.

Choosing Douban Movie as the target network, we firstly set the training ratio r_t as 0.2 or 1.0, and partition the collected movie items in the target network $\mathcal{G}^{(a)}$ with 5 folds cross validation: one fold as $\mathcal{I}_x$ which denotes the set of the newly imported movie items, the rest 4 folds to form $\mathcal{I}_y$ which is the set of the old items that already exist in $\mathcal{G}^{(1)}$ for a time. And use the observed ratings relate to the items in $\mathcal{I}_x$ to form the test set T_E. Secondly, we randomly sample r_t of the ratings in $\mathcal{G}^{(a)}$ that relate to the items in $\mathcal{I}_y$ to form the set T_T. Thirdly, we combine T_T with the set of all the observed ratings in the source network $\mathcal{G}^{(b)}$ to form the training set T_R. Here, $r_t = 0.2$ denotes that the target network is very new and only contains a small amount of ratings, while $r_t = 1.0$ denotes that the target network is old enough and has a certain amount of ratings.

Figure 5 shows the impacts of α and β on MAE and RMSE in *CHRS* model. We can find that with the same r_t, the performances of *CHRS* on MAE and RMSE have very similar trend. Moreover, the values of α and β 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. And the results prove that α and β should be set with moderate values to make *CHRS* perform well: for $r_t = 0.2$, *CHRS* can achieve its best performance when $\alpha = 500$ and $\beta = 0.1$; while for $r_t = 1.0$, *CHRS* can achieve its best performance when $\alpha = 1000$ and $\beta = 0.5$. And for very small α and β, *CHRS* will degrade to the traditional *LMF* model, which makes its MAE and RSME increase to higher and stable values (i.e., bad performance). For very large α and β, the item similarity information and the process of item information transfer will dominate model learning process, which also makes the MAE and RSME values of *CHRS* increase.

VI. CONCLUSIONS

In this paper, we propose a *Cross-HIN Recommendation System(CHRS)*, which uses a two-step approach to integrate the auxiliary information in both of the source and target networks for recommendation. By utilizing the rich information from meta-paths, *CHRS* is able to calculate the movie similarities from multiple types of relation information. These calculated item similarities are used by the *Item Similarity Regulation* terms of *CHRS* to improve recommendation performances. And basing on item anchor links, *CHRS* adopt an unidirectional way to transfer item latent information from the source network to the target network. During the transfer process, a domain adaptation matrix is used to overcome the domain difference problem. In this way, our *CHRS* can solve the ''cold start'' problem, which can hardly well solved by many existing recommendation methods. We conduct experiments to compare our *CHRS* method with several widely employed or state-of-the-art recommendation methods, and the experimental results reflect that our method outperforms the other base-line methods in different ''cold start'' and ''semi-cold start'' circumstances. We also design experiments to study the effects of the parameters of *CHRS* on its performances. The results show that it is desirable to design clever strategy to learn the parameters for the combination of item similarities and cross-network information to further improve recommendation performances.

REFERENCES

[1] Y. Sun, R. Barber, M. Gupta, J. Han, and C. C. Aggarwal, ''Co-author relationship prediction in heterogeneous bibliographic networks,'' in *Proc. Int. Conf. Adv. Social Netw. Anal. Mining*, Jul. 2011, pp. 121–128.

[2] W. Lin, X. Kong, P. S. Yu, Q. Wu, Y. Jia, and C. Li, ''Community detection in incomplete information networks,'' in *Proc. 21st Int. Conf. World Wide Web*, 2012, pp. 341–350.

[3] J. Wu, L. Chen, Q. Yu, P. Han, and Z. Wu, ''Trust-aware media recommendation in heterogeneous social networks,'' *World Wide Web*, vol. 18, no. 1, pp. 139–157, 2015.

[4] W. Pan, ''A survey of transfer learning for collaborative recommendation with auxiliary data,'' *Neurocomputing*, vol. 177, pp. 447–453, Feb. 2016.

[5] X. Yu, X. Ren, Q. Gu, Y. Sun, and J. Han, ''Collaborative filtering with entity similarity regularization in heterogeneous information networks,'' in *Proc. IJCAI HINA*, 2013, pp. 1–6.

[6] C. Shi, J. Liu, F. Zhuang, P. S. Yu, and B. Wu, ''Integrating heterogeneous information via flexible regularization framework for recommendation,'' *Knowl. Inf. Syst.*, vol. 49, no. 3, pp. 1–25, 2015.

[7] C. Luo, W. Pang, Z. Wang, and C. Lin, ''Hete-CF: Social-based collaborative filtering recommendation using heterogeneous relations,'' in *Proc. IEEE Int. Conf. Data Mining*, Sep. 2014, pp. 917–922.

[8] A. S. Tewari, A. Kumar, and A. G. Barman, ''Book recommendation system based on combine features of content based filtering, collaborative filtering and association rule mining,'' in *Proc. IEEE Int. Adv. Comput. Conf.*, Apr. 2014, pp. 500–503.

[9] W. Pan, E. W. Xiang, N. N. Liu, and Q. Yang, ''Transfer learning in collaborative filtering for sparsity reduction,'' in *Proc. AAAI*, vol. 10. 2010, pp. 230–235.

[10] M. Yan, J. Sang, C. Xu, and M. S. Hossain, ''A unified video recommendation by cross-network user modeling,'' *ACM Trans. Multimedia Comput. Commun. Appl. (TOMM)*, vol. 12, no. 4, p. 53, 2016.

[11] M. Yan, J. Sang, and C. Xu, ''Unified YouTube video recommendation via cross-network collaboration,'' in *Proc. 5th ACM Int. Conf. Multimedia Retr.*, 2015, pp. 19–26.

[12] W. Chen, W. Hsu, and M. L. Lee, ''Making recommendations from multiple domains,'' in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 892–900.

[13] Y. Shi, M. Larson, and A. Hanjalic, ''Tags as bridges between domains: Improving recommendation with tag-induced cross-domain collaborative filtering'' in *Proc. User Modeling Adaption Pers. Int. Conf. (UMAP)*, Girona, Spain, Jul. 2011, pp. 305–316.

[14] K. Weiss, T. M. Khoshgoftaar, and D. Wang, ''A survey of transfer learning,'' *J. Big Data*, vol. 3, no. 1, pp. 1–40, 2016.

[15] J. Zhang, P. S. Yu, and Z.-H. Zhou, ''Meta-path based multi-network collective link prediction,'' in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 1286–1295.

[16] X. Kong, J. Zhang, and P. S. Yu, ''Inferring anchor links across multiple heterogeneous social networks,'' in *Proc. 22nd ACM Int. Conf. Inf. Knowl. Manage.*, 2013, pp. 179–188.

[17] I. Bhattacharya and L. Getoor, ''Collective entity resolution in relational data,'' *ACM Trans. Knowl. Discovery From Data (TKDD)*, vol. 1, no. 1, p. 5, 2007.

[18] M. Tsikerdekis and S. Zeadally, ''Multiple account identity deception detection in social media using nonverbal behavior,'' *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 8, pp. 1311–1321, Oct. 2014.

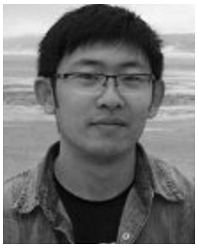
[19] S. Liu, S. Wang, F. Zhu, J. Zhang, and R. Krishnan, ''HYDRA: Large-scale social identity linkage via heterogeneous behavior modeling,'' in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2014, pp. 51–62.

[20] Y. Zhang, J. Tang, Z. Yang, J. Pei, and P. S. Yu, ''COSNET: Connecting heterogeneous social networks with local and global consistency,'' in *Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Sep. 2015, pp. 1485–1494.

[21] R. Zafarani and H. Liu, "Connecting users across social media sites: A behavioral-modeling approach," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 41–49.
[22] C.-T. Lu, H.-H. Shuai, and P. S. Yu, "Identifying your customers in social networks," in *Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manage.*, 2014, pp. 391–400.
[23] D. Liu and Q. Y. Wu, "Cross-platform user profile matching in online social networks," *Appl. Mech. Mater.*, vol. 380, pp. 1955–1958, Sep. 2013.
[24] J. Zhang and S. Y. Philip, "Multiple anonymized social networks alignment," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Sep. 2015, pp. 599–608.
[25] J. Zhang and P. S. Yu, "PCT: Partial co-alignment of social networks," in *Proc. 25th Int. Conf. World Wide Web*, Apr. 2016, pp. 749–759.
[26] J. Zhang, X. Kong, and S. Y. Philip, "Predicting social links for new users across aligned heterogeneous social networks," in *Proc. IEEE 13th Int. Conf. Data Mining*, May 2013, pp. 1289–1294.
[27] J. Zhang and S. Y. Philip, "Integrated anchor and social link predictions across social networks," in *Proc. 24th Int. Conf. Artif. Intell.*, 2015, pp. 2125–2131.
[28] L. Xu, C. Jiang, J. Wang, J. Yuan, and Y. Ren, "Information security in big data: Privacy and data mining," *IEEE Access*, vol. 2, pp. 1149–1176, Oct. 2014.
[29] W. Pan, N. N. Liu, E. W. Xiang, and Q. Yang, "Transfer learning to predict missing ratings via heterogeneous user feedbacks," in *Proc. IJCAI Int. Joint Conf. Artif. Intell.*, vol. 22. 2011, p. 2318.
[30] C. Shi, X. Kong, Y. Huang, P. S. Yu, and B. Wu, "HeteSim: A general framework for relevance measure in heterogeneous networks," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 10, pp. 2479–2492, Sep. 2013.
[31] C. T. Lu, S. Xie, W. Shao, L. He, and P. S. Yu, "Item recommendation for emerging online businesses," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 1–7.
[32] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web*, 2001, pp. 285–295.
[33] X. Su and T. M. Khoshgoftaar, *A Survey of Collaborative Filtering Techniques*. London, U.K.: Hindawi Publishing Corp., 2009.
[34] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Application of dimensionality reduction in recommender system—A case study," in *Proc. ACM Webkdd Workshop*, 2000.
[35] N. Srebro and T. Jaakkola, "Weighted low-rank approximations," in *Proc. Int. Conf. Mach. Learn.*, 2003, pp. 720–727.
[36] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proc. 4th Int. Conf. Web Search Web Data Mining (WSDM)*, Hong Kong, Feb. 2011, pp. 287–296.
[37] J. Liu, Y. Jiang, Z. Li, X. Zhang, and H. Lu, "Domain-sensitive recommendation with user-item subgroup analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 4, pp. 939–950, Apr. 2016.
[38] W. Wang, H. Yin, S. Sadiq, L. Chen, M. Xie, and X. Zhou, "Spore: A sequential personalized spatial item recommender system," in *Proc. IEEE Int. Conf. Data Eng.*, Sep. 2016, pp. 954–965.
[39] G. Guo, J. Zhang, and N. Yorke-Smith, "A novel recommendation model regularized with user trust and item ratings," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 7, pp. 1607–1620, Sep. 2016.
[40] Z. Xu, F. Zhang, W. Wang, H. Liu, and X. Kong, "Exploiting trust and usage context for cross-domain recommendation," *IEEE Access*, vol. 4, pp. 2398–2407, 2016.
[41] T. Li, A. Liu, and C. Huang, "A similarity scenario-based recommendation model with small disturbances for unknown items in social networks," *IEEE Access*, vol. 4, pp. 9251–9272, 2016.
[42] W. Zhang and J. Wang, "A collective Bayesian Poisson factorization model for cold-start local event recommendation," in *Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 1455–1464.
[43] W. X. Zhao, S. Li, Y. He, E. Y. Chang, J.-R. Wen, and X. Li, "Connecting social media to e-commerce: Cold-start product recommendation using microblogging information," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 5, pp. 1147–1159, May 2016.
[44] R. Zafarani and H. Liu, "Connecting corresponding identities across communities," in *Proc. ICWSM*, vol. 9. 2009, pp. 354–357.
[45] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "PathSim: Meta path-based top-k similarity search in heterogeneous information networks," in *Proc. Vldb Endowment*, vol. 4, no. 11, pp. 992–1003, 2011.
[46] N. Lao and W. W. Cohen, "Fast query execution for retrieval models based on path-constrained random walks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Washington, DC, USA, Jul. 2010, pp. 881–888.
[47] F. Nie, H. Huang, X. Cai, and C. Ding, "Efficient and robust feature selection via joint $\ell 2,1$-norms minimization," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2010, pp. 1813–1821.
[48] C. T. Kelley, *Iterative Methods for Optimization*, vol. 18. Philadelphia, PA, USA: SIAM, 1999.

JUNXING ZHU received the M.Sc. degree from the College of Computer, National University of Defense Technology, China, in 2013, where he is currently pursuing the Ph.D. degree. His research interests include social network analysis, data mining, and spam filtering.

JIAWEI ZHANG received the bachelor's degree in computer science from Nanjing University, China, in 2012, and the Ph.D. degree in computer science from the University of Illinois at Chicago, USA, in 2017. He has been an Assistant Professor with the Department of Computer Science, Florida State University, Tallahassee, FL, USA, since 2017. His main research areas are data mining and machine learning, especially multiple aligned social networks studies.

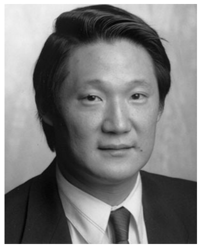


CHENWEI ZHANG received the B.S. degree in computer science and technology from Southwest University, China, in 2014. He is currently pursuing the Ph.D. degree with the Department of Computer Science, University of Illinois at Chicago. His research interests lie in the fields of data mining and deep learning. In particular, he is interested in text mining and mining structured information from heterogeneous information sources.

QUANYUAN WU received the Ph.D. degree from Fudan University, China, in 1965. He was appointed as a Professor with the College of Computer, National University of Defense Technology, China, in 1986. He has authored over 80 research papers in international journals and conferences. His research interests include artificial intelligence, cloud computing, middleware, and big data. He was awarded as the National Outstanding Mid-Aged Expert by the Chinese government in 1990, and began to enjoy the special subsidy of the State Council since 1991.

YAN JIA was born in 1960. She is currently a Professor and a Ph.D. Supervisor with the College of Computer, National University of Defense Technology, China. Her main research interests include database, social network analysis, and data mining.

BIN ZHOU is currently a Professor of Computer Science with the National University of Defense Technology, Changsha, China. His main research interests include Web text mining, online social network (OSN) analysis, and big data processing. He has published over 100 research papers (over 20 were SCI indexed and over 60 EI indexed) on these topics. He also received several academic rewards, including two national science and technology progress awards (second class), four science and technology progress awards of Hunan Province (first class twice and second class twice). Recently, he has been involved in several international conference program/organization committees relating to OSN and big data processing, such as APWeb2014, ASONAM2014, and CCF Bigdata2014.



PHILIP S. YU (F'93) received the B.S. degree in electrical engineering from National Taiwan University, the M.S. and Ph.D. degrees in EE from Stanford University, and the M.B.A. degree from New York University. He was with IBM, where he was the Manager with the Software Tools and Techniques Group, Watson Research Center. He is currently a Distinguished Professor of Computer Science with the University of Illinois at Chicago and also holds the Wexler Chair in information technology. He has published over 970 papers in refereed journals and conferences. He holds or has applied for over 300 U.S. patents. His research interest is on big data, including data mining, data stream, database, and privacy. He is a fellow of the ACM. He is on the Steering Committee of the ACM Conference on Information and Knowledge Management and was a member of the Steering Committee of the IEEE Data Engineering and the IEEE Conference on Data Mining. He received the ACM SIGKDD 2016 Innovation Award for his influential research and scientific contributions on mining, fusion, and anonymization of big data, the IEEE Computer Society's 2013 Technical Achievement Award for pioneering and fundamentally innovative contributions to the scalable indexing, querying, searching, mining, and anonymization of big data, and the Research Contributions Award from the IEEE International Conference on Data Mining (ICDM) in 2003 for his pioneering contributions to the field of data mining. He also received the ICDM 2013 10-year Highest-Impact Paper Award and the EDBT Test of Time Award in 2014. He has received several IBM honors, including two IBM Outstanding Innovation Awards, an Outstanding Technical Achievement Award, two Research Division Awards, and the 94th plateau of Invention Achievement Awards. He was an IBM Master Inventor. He was the Editor-in-Chief of the IEEE Transactions on Knowledge and Data Engineering from 2001 to 2004. He is the Editor-in-Chief of the *ACM Transactions on Knowledge Discovery from Data*.