Multimedia Tools and Applications (2021) 80:24245–24270
https://doi.org/10.1007/s11042-021-10834-y

Enhanced knowledge transfer for collaborative filtering with multi-source heterogeneous feedbacks

Hongwei Zhang[1,2] · Xiangwei Kong[3] · Yujia Zhang[4]

Received: 25 July 2020 / Revised: 22 February 2021 / Accepted: 10 March 2021 /
Published online: 3 April 2021

Abstract

Collaborative filtering (CF) is a widely used method in recommender systems due to its simplicity and efficiency. But most existing CF methods suffer from data scarcity, which arises from the situation with only a limited number of interactions between users and items. One solution to address is how to introduce Transfer Learning (TL)-based CF methods with heterogeneous feedbacks to deal with multi-source and heterogeneous data, such as rating vs. clicks or rating vs. purchase. However, in some applications, extremely sparse (i.e., sparsity level $\leq 0.1\%$) target data could cause under-transfer and negative transfer. To address the above issue, we propose an Enhanced Knowledge Transfer for Collaborative Filtering with Multi-Source Heterogeneous Feedbacks (EKT). Specifically, we first propose a weighted collective matrix tri-factorization framework. The proposed framework constrains the auxiliary data and the target data to share the same latent factors of users and items as well as partial cluster-level user-item rating pattern in order to enhance knowledge transfer and alleviate the under-transfer issue. Then, to alleviate the negative transfer issue, we integrate the graph co-regularization terms into a proposed framework, which contains the neighborhood structure information of users and items. At last, we simultaneously minimize the objective function of EKT, which consists of weighted collective matrix tri-factorization and the graph co-regularization of user and item graphs. Since the EKT framework is a non-convex optimization problem, we use an alternating optimization procedure to solve it and further prove its convergence. The experimental results on two benchmark datasets show that our proposed EKT method performs better than other baseline methods at almost all sparsity levels except for the denser case of 1% on ML10M and Netflix.

Keywords Transfer learning · Collaborative filtering · Sparsity · Heterogeneous feedbacks

1 Introduction

Recommender systems are one of the most widely used applications by many on-line services, such as E-commerce platforms, news portals, advertising, social media sites, etc [1]

✉ Xiangwei Kong
kongxiangwei@zju.edu.cn

Extended author information available on the last page of the article.

for information overload problem. Generally speaking, the mainly used recommendation techniques are roughly classified as content-based and collaborative filtering-based. Since the Collaborative Filtering (CF) method has no content restriction, it is more widely used in both literature and industry [7, 8, 13, 36, 42]. In recommender systems, CF aims to predict the missing ratings for an item or a user based on the collected ratings from similar items or like-minded users. [9, 14, 32, 40]. CF has become one of the most popular methods in recommender systems at present due to its simpleness and high-efficient [39, 41], but it also faces major bottlenecks such as data sparsity [18]. The so-called data sparsity refers to the fact that too few data is observed in the user-item rating matrix, which makes the recommendation model suffers from overfitting and leads to low-quality predictions [30].

To cope with the sparsity issue, numerous improved CF methods have been proposed. Among them, Matrix Factorization (MF) gains great success during the past ten plus years. For example, Srebro, etc. [37] presented the Maximum Margin Matrix Factorization (MMMF) method, whose goal is to learn a complete observed matrix with the minimum trace norm by maximizing the prediction margin to approximate the target preference matrix. Zhang et al. [45] presented a weighted nonnegative matrix factorization (WNMF) method that decomposes the observed user-item rating matrix into two low-dimensional nonnegative matrices, and then uses their product to predict the unobserved user-item rating. Gu et al. [11] extended the WNMF method by incorporating user and item graphs. Chen et al. [6] applied Orthogonal Nonnegative Matrix Tri-Factorization (ONMTF) for collaborative filtering to alleviate the sparsity problem. Mnih et al. [24] proposed a probability matrix factorization (PMF) method, which uses a probability model with Gaussian observation noise, with the goal of maximizing the conditional distribution on the observation rating. The recommendation accuracy of these MF methods still largely depends on the target rating matrix. However, when the observed target rating matrix is very sparse, the recommendation performance would be degraded seriously.

Recently, transfer learning (TL) [25] methods have been introduced into collaborative filtering for solving the data sparsity issue [15, 22, 35, 46]. The kernel thought behind it is to extract the common latent knowledge from some dense auxiliary data via the latent factorization model, then transfer it to sparse target data [3, 33]. The existing TL-based CF methods mostly focus on selecting cluster-level codebooks [18, 19], or the latent tastes of users/latent features of items [23, 33, 35] as common knowledge for transfer. They are often limited to the transfer of homogeneous user feedbacks. However, heterogeneous user feedbacks are more common in reality. To this end, several TL-based CF methods using heterogeneous user feedback have been proposed to solve the data sparsity issue. For example, Pan et al. [27, 30, 31] explore how to utilize dense binary preference data (“like” or “dislike”) in the auxiliary domain to aid recommendations in the target domain with numerical rating data (5-star rating). However, in practical scenarios, the target rating matrix is often extremely sparse (e.g., sparsity level ≤ 0.1), at which point these methods are likely to encounter the following two limitations. First, the under-transfer issue is likely to occur as the extremely sparse target data requires more knowledge to be transferred from the auxiliary data. The so-called under-transfer means that the useful knowledge in the auxiliary data is not fully transferred to the target data. Second, the latent factors extracted from the auxiliary data are likely to transfer the negative information to the target data, resulting in the negative transfer issue.

In this paper, we investigate how to transfer knowledge effectively from dense auxiliary binary rating data to extremely sparse target numerical rating data to improve the prediction accuracy of the recommender system. To address the problem mentioned above, we proposed a new TL-based CF method, referred to as Enhanced Knowledge Transfer for

Collaborative Filtering with Multi-Source Heterogeneous Feedbacks (EKT). First, we use the proposed weighted collective matrix tri-factorization framework to extract the common latent factors of users and items as well as the common partial cluster-level rating pattern, through which more knowledge can be transferred between domains. Second, we incorporate the graph co-regularization of user and item graphs into the proposed weighted collective matrix tri-factorization, which will preserve the intrinsic geometric structure in each domain, thus alleviating the negative transfer issue. Simultaneously, when the target data is extremely sparse, the neighborhood structure information of the auxiliary data can be transferred to the target data, further enhancing the transfer of knowledge. The main contributions of this paper are summarized as follows:

- We propose a new TL-based CF framework, which integrates weighted collective matrix tri-factorization and graph co-regularization of user and item graphs in a unified framework. The proposed framework can alleviate the under-transfer and negative transfer issues that may be caused by extremely sparse target data.
- For the proposed framework, we propose an alternative optimization procedure and further prove its convergence.
- On two benchmark data sets, we demonstrate the effectiveness of proposed EKT method at a variety of sparsity levels of $0.01\% \sim 1\%$, and the proposed EKT method shows better performance compared to several state-of-the-art baseline methods when the sparsity level is less than 1%.

The rest of the paper is organized as follows. We first review some related works briefly in Section 2, then describe the proposed EKT method in detail in Section 3. After that we show experiments conducted on two real-world data sets to verify the effectiveness of the proposed EKT mehod in Section 4. Finally, we give the conclusion and directions for future study in Section 5. The notations used through the paper are listed in Table 1.

Table 1 Notations

Notation	Description
M	#target (auxiliary) users
N	#target (auxiliary) items
$\mathcal{D}_\tau$	domain τ, $\tau \in \{t, a\}$
d_1	#latent tastes of users
d_2	#latent features of items
c	#shared cluster-level user-item rating pattern
X_τ	$M \times N$ rating matrix of D_τ
Y_τ	$M \times N$ indicator matrix of D_τ
U_τ	d_1 latent tastes of users in D_τ
V_τ	d_2 latent features of items in D_τ
B_τ	$d_1 \times d_2$ cluster-level rating pattern matrix of X_τ
S	shared cluster-level user-item rating pattern matrix
S_τ	specific cluster-level rating pattern matrix of X_τ
λ	tradeoff parameter
α_u, α_v	graph regularization parameter in $\mathcal{D}_t$
β_u, β_v	graph regularization parameter in $\mathcal{D}_a$
$\alpha_s, \beta_s, \gamma_s$	regularization parameter of rating pattern matrix
θ_u, θ_v	regularization parameter of latent factors
K	#iterations

2 Related work

This paper focuses on improving the recommendation accuracy of collaborative filtering methods when rating data is extremely sparse. In this section, we will discuss the related works.

Collaborative filtering is a simple and effective recommendation method, but when the target data is very sparse, it is easy to obtain degraded prediction results [38]. In recent years, transfer learning has been introduced for solving data sparsity problems in collaborative filtering [12, 20, 33], which transfer knowledge from auxiliary data to target data. From the content of the transfer, most of the existing transfer learning methods in collaborative filtering are mainly considered from two perspectives. One is the cluster-level rating pattern transfer, which transfers the cluster-level rating pattern from the auxiliary data to the target data. For example, Bin Li et al. proposed Codebook Based Transfer (CBT) model [18] to transfer knowledge of cluster-level rating behavior from auxiliary data of movies to target data of books. A further extension of CBT is known as the Rating Matrix Generation Model (RMGM) [19], which relaxes the hard membership constraint on user/item groups to soft membership. Gao et al. [10] extended Bin Li at el.'s methods and proposed cluster-level based Latent Factor Model (CLFM) that achieves knowledge transfer by sharing partial cluster-level rating patterns across multiple domains. Qian Zhang et al. [44] believe that the CBT method cannot ensure that the knowledge extracted from the auxiliary domain is consistent with the target domain. To this end, they presented a cross-domain recommender system with Consistent Information Transfer (CIT) method [44], which adopts a domain adaptation strategy to make the distribution of latent factors in two domains as close as possible, and then perform a codebook transfer. Cluster-level rating pattern transfer mostly targets scenarios where there is no entity (user or item) correspondence between target data and auxiliary data. Because of the size limitation of the cluster-level rating pattern, it cannot transfer enough useful knowledge when the observed target data is quite sparse.

The other is latent factors transfer, which is to transfer user/item latent factors from the auxiliary data to the target data. CMF [35] is a multi-task learning method, which jointly factorizes the target rating matrix and item-side content matrix while sharing the same latent factors of items. Similarly, Hao Ma et al. [23] proposed SoRec method, which alternatively factorizes the target rating matrix and a user-side social network matrix while sharing the same latent factor of users. Pan et al. proposed Coordinate System Transfer (CST) model [30] and Transfer by Collective Factorization (TCF) [27, 31] to transfer the latent factors of users and items from auxiliary dense binary rating data to sparse target numerical rating data. But the two models do not take into account the neighborhood information between users and between items. Shi et al. [33] proposed twin bridge transfer learning (TBT), which transfers knowledge from dense auxiliary data to sparse target data by using latent factors and similarity graphs as twin bridges. Pan et al. [28] extended the CMF method and proposed interaction-rich transfer by collective factorization (iTCF), which not only constrains the sharing of the same item latent factors between target data and auxiliary data, but also requires information interaction between user latent factors in the two domains. Pan et al. [29] further extended the iTCF method and proposed Transfer by Mixed Factorization (TMF) approach, which introduces a virtual user profile to model the user's implicit preference based on the iTCF method. The virtual user profile is composed of latent factors of items that the user likes and dislikes in the target domain. Zhao et al. [46] relaxed the assumption that there is an adequate set of entity correspondences across domains and employed an active learning principle to construct entity correspondences across domains. Then the actively constructed entity correspondences are plugged into a general transferred CF model to improve recommendation performance. Zhang et al. [43] proposed a cross-domain recommender system based on kernel-induced (KerKT) in a scenario where there are only partially overlapping entities between domains. kerKL exploits domain adaptation technology to adjust the feature spaces between overlapping entities, and adopts kernel diffusion completion to correlate non-overlapping entities between domains.

Pan et al. [26] gave a comprehensive survey of transfer learning for collaborative recommendation with auxiliary data, which mainly consider the CRAD (Collaborative Recommendation with Auxiliary Data) problem from a transfer learning view, and then discuss three knowledge transfer strategies for collaborative recommendation with different types of auxiliary data. Most existing TL-based CF technologies can be summarized into these three strategies, namely adaptive knowledge transfer, collective knowledge transfer, and integrated knowledge transfer. According to the survey of Pan et al. [26], our proposed EKT method belongs to collective knowledge transfer, which aims to jointly learn the shared knowledge and unshared effect of the target data and auxiliary data simultaneously. The advantage of collective knowledge transfer is that it can perform bi-directed knowledge transfer, thus it has richer interactions similar to multi-task learning. Most of the previously mentioned methods mainly focus on either latent factor transfer or cluster-level rating pattern transfer. Different from the previously work, in this paper, we try to jointly transfer the latent factors and the partial cluster-level rating patterns simultaneously, to enhance more knowledge to transfer. In addition, to alleviate the possible negative transfer issue, we use the graph co-regularization of user and item graphs to refine the latent factors of user and item to preserve the intrinsic geometric structure.

3 Enhanced transfer learning for collaborative filtering with multi-source heterogeneous feedbacks

In this section, we first define the problem setting, then propose Enhanced Knowledge Transfer for collaborative filtering with Multi-Source Heterogeneous Feedbacks (EKT) framework. Finally, we will show the optimization process of the proposed EKT method and analyze its convergence.

3.1 Problem formulation

In the target data, suppose there are a user-item numerical rating matrix $X_t = [(x_t)_{ij}]_{M\times N} \in \{1, 2, 3, 4, 5, ?\}^{M\times N}$ ("?" denotes a missing value). $Y_t = [(y_t)_{ij}]_{M\times N} \in \{0, 1\}^{M\times N}$ is the indicator matrix, $(y_t)_{ij} = 1$ if user i has rated item j, and $(y_t)_{ij} = 0$ otherwise. Similarly, in the auxiliary data, there are a user-item binary rating matrix $X_a = [(x_a)_{ij}]_{M\times N} \in \{0, 1, ?\}^{M\times N}$, where "0" and "1" represent the observed "like" value and "dislike" value, respectively. The question mark "?" represents the missing value. $Y_a = [(y_a)_{ij}]_{M\times N} \in \{0, 1\}^{M\times N}$ is the corresponding indicator matrix. It is assumed here that users and items of X_t and X_a are aligned, that is, one-to-one mapping. We aim to predict the missing values in the target rating matrix X_t by transferring the rating knowledge in the auxiliary rating matrix X_a. According to Li's description of "domain" [17], X_t and X_a can be seen as coming from different data domains. Denote $\mathcal{D}_\tau$ as the τ domain, where $\tau \in \{t, a\}$ is the domain index. Therefore, we can think of it as a cross-domain recommendation problem.

3.2 EKT method

The proposed EKT framework incorporates two learning objectives into a uniform optimization problem: weighted collective matrix tri-factorization and the graph co-regularization of user and item graphs. The graph model is shown in Fig. 1.

3.2.1 Weighted collective matrix Tri-factorization

First, we proposed a Weighted Collective Matrix Tri-Factorization framework (WCMTF) by extending the Collective Matrix Factorization (CMF) and Weighted Nonnegative Matrix Tri-Factorization (WNMTF). Then we use WCMTF to extract the latent factors and cluster-level rating pattern, which can narrow the data distribution between domains and transfer more latent knowledge from the auxiliary domain to the target domain.

Given the target numerical rating matrix X_t and the auxiliary binary rating matrix X_a, we can extract the latent factors of each rating matrix via Weighted Nonnegative Matrix Tri-Factorization (WNTMF) [11]. In WNTMF, the nonnegative user-item rating matrix X_τ can be decomposed into three low-rank matrices, U_τ, B_τ and V_τ, such that the

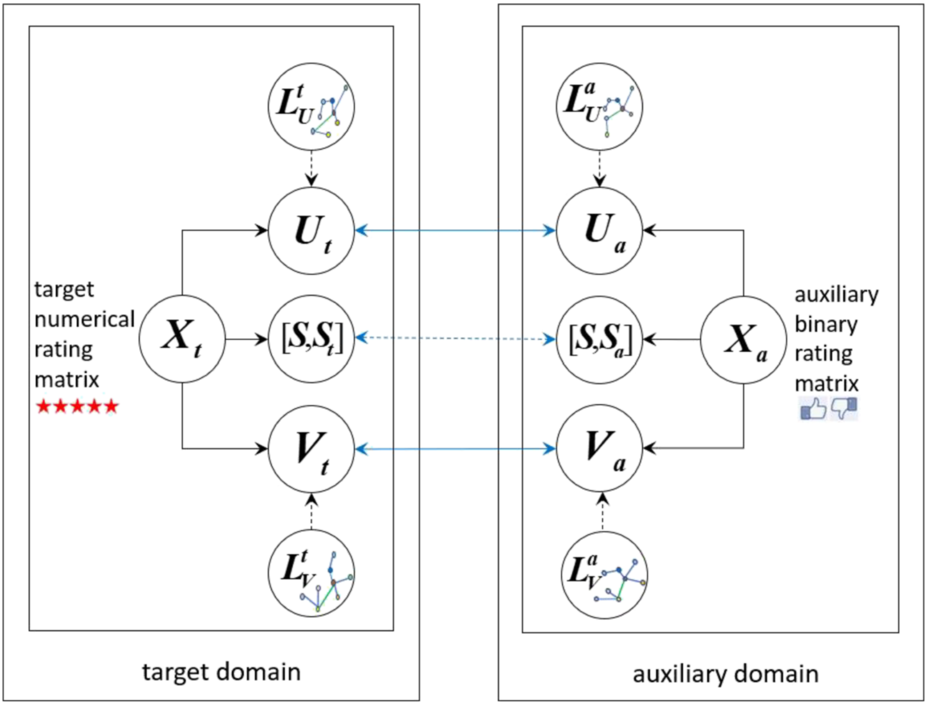


Fig. 1 Graphical model of Enhanced Transfer Learning for Collaborative Filtering with Multi-Source Heterogeneous Feedbacks. The target numerical rating matrix X_t and the auxiliary binary rating matrix X_a are jointly decomposed. Note that the blue solid double arrow indicates that the two domains share the same user/item's latent factor. The blue dashed double arrow indicates that the two domains share the partial cluster-level user-item rating pattern. The black dashed arrow indicates that the graph regularization constraint is imposed on the latent factor of the user/item

reconstruction error of matrix X_τ is minimized. WNMTF is equivalent to the following optimization problem:

$$\min_{U_\tau, B_\tau, V_\tau \geq 0} \left\| Y_\tau \odot \left(X_\tau - U_\tau B_\tau V_\tau^T \right) \right\|_F^2, \tag{1}$$

where $\odot$ represents the element-wise product of matrices, and Y_τ is the indicator matrix denoting whether the rating in X_τ is observed or not. $U_\tau = [u_{\cdot 1}^\tau, u_{\cdot 2}^\tau, \cdots, u_{\cdot d_1}^\tau] \in \mathbb{R}^{M \times d_1}$ represents the latent factor matrix of users, with each $u_{i\cdot}^\tau$ denoting the latent taste of user i. $V_\tau = [v_{\cdot 1}^\tau, v_{\cdot 2}^\tau, \cdots, v_{\cdot d_2}^\tau] \in \mathbb{R}^{N \times d_2}$ represents the latent factor matrix of items, with each $v_{i\cdot}^\tau$ denoting the latent feature of item i. $B_\tau \in \mathbb{R}^{d_1 \times d_2}$ is the cluster-level user-item rating pattern representing the association between U_τ and V_τ, $\tau \in \{t, a\}$.

Since the users and items of the target numerical rating matrix and the auxiliary binary rating matrix are aligned in our hypothesis, they share potential users tastes and item features. We may improve prediction accuracy by sharing the common latent factors of users and items underlying these two rating data. Similar to CMF [35], we extend the basic WNMTF to decompose two relevant matrices simultaneously, resulting in a weighted collective matrix tri-factorization

$$\begin{aligned} \min_{U_t, U_a, B_t, B_a, V_t, V_a \geq 0} & \left\| Y_t \odot \left(X_t - U_t B_t V_t^T \right) \right\|_F^2 + \lambda \left\| Y_a \odot \left(X_a - U_a B_a V_a^T \right) \right\|_F^2 \\ s.t. \;\; & U_t \equiv U_a \equiv U, V_t \equiv V_a \equiv V, \end{aligned} \tag{2}$$

where $\lambda > 0$ is a trade-off parameter used to balance the target data with the auxiliary data. The above optimization problem can be further simplified into the following form

$$\min_{U, B_t, B_a, V \geq 0} \left\| Y_t \odot \left(X_t - U B_t V^T \right) \right\|_F^2 + \lambda \left\| Y_a \odot \left(X_a - U B_a V^T \right) \right\|_F^2. \tag{3}$$

Without considering the regularization terms, the form of formula (3) is similar to the TCF framework [27, 31]. Note, however, that there are no non-negative constraints on the variables U, B_a, B_t, and V in the TCF. The TCF method and formula (3) both only transfer knowledge by sharing the same latent factors of users and items. However, when the target data is extremely sparse, it is desirable to transfer more common knowledge from the auxiliary data to alleviate the sparsity of the target data. Therefore, we believe that the knowledge transfer of them is likely to be insufficient. Gao et al. [10] proposed CLFM model, which can not only learn the common rating pattern shared cross-domain, but also learn the domain-specific rating pattern of users in each domain. The rating pattern matrix can be considered as the probability that the user group rates the corresponding item cluster. Inspired by this, we consider that the partial cluster-level user-item rating pattern can be used as a new bridge for knowledge transfer. Although the rating form of the target data and the auxiliary data are heterogeneous, i.e., $\{1, 2, 3, 4, 5, ?\}$ and $\{0, 1, ?\}$, the rating form of the pre-processed target data and the auxiliary data is partially aligned, i.e., $\{0, 0.25, 0.5, 0.75, 1, ?\}$ and $\{0, 1, ?\}$. Therefore, we can further assume that the preprocessed target data and auxiliary data share partial cluster-level user-item rating pattern.

To this end, we assume that the cluster-level rating patterns B_t and B_s in the formula (3) can be expressed as $[S, S_t]$ and $[S, S_a]$, respectively. Formally, the collective matrix tri-factorization framework in (3) can be re-represented as follows

$$\min_{U, S, S_t, S_a, V \geq 0} \left\| Y_t \odot \left(X_t - U[S, S_t] V^T \right) \right\|_F^2 + \lambda \left\| Y_a \odot \left(X_a - U[S, S_a] V^T \right) \right\|_F^2, \tag{4}$$

where U and V denote the shared user latent factor matrix and item latent factor matrix,respectively. S denotes the shared user-item rating pattern matrix. S_t and S_a denote the specific rating pattern matrix of target data and auxiliary data, respectively.

In formula (4), we require that the latent factor matrices of users and items in both domains are exactly the same. However, although the target domain and the auxiliary domain are related, the latent user tastes and item features from two domains can still be somewhat different due to the domain specific contexture. Therefore, here we relax the constraint that the latent factors for users and items in both domains are exactly the same, and only require that they are similar, which can be achieved by adding the regularization terms $\|U_t - U_a\|_F^2$ and $\|V_t - V_a\|_F^2$ in the framework (4). The objective function in framework (4) can be further expressed as

$$\begin{aligned}\min_{U_t,U_a,V_t,V_a,S,S_t,S_a\geq 0} \quad & \left\| Y_t \odot (X_t - U_t[S, S_t]{V_t}^T) \right\|_F^2 \\ & + \lambda \left\| Y_a \odot (X_a - U_a[S, S_a]{V_a}^T) \right\|_F^2 \\ & + \theta_u \left\| U_t - U_a \right\|_F^2 + \theta_v \left\| V_t - V_a \right\|_F^2 , \end{aligned} \tag{5}$$

where U_t and V_t denote the latent factor matrix for users and items from target data, respectively. U_a and V_a denote the latent factor matrix for users and items from auxiliary data, respectively. θ_u, θ_v is the tradeoff parameter, representing the confidence in the auxiliary data. The illustration of our proposed WCMTF framework can be found in Fig. 2.

In the formula (5), the useful knowledge from auxiliary data can be adequately transferred to the target data by sharing the user latent factors, item latent factors and partial cluster-level user-item rating pattern. However, more knowledge transfer may also lead to more serious negative transfer issue, that is, harmful information in auxiliary data is transferred to target data. Especially when the target data is extremely sparse, negative transfer is more likely to occur. To do this, we need to find ways to alleviate the possible negative transfer issue.

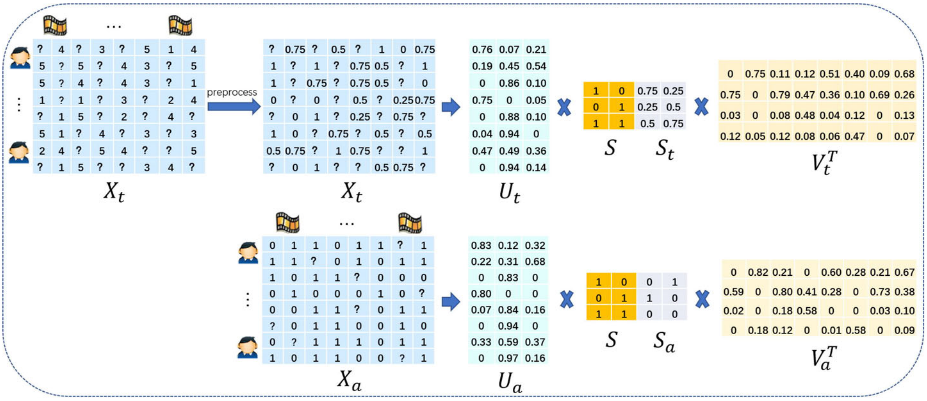


Fig. 2 Illustration of our proposed WCMTF framework

3.2.2 Graph co-regularization of user and item graphs

We adopt the co-regularization of user and item graphs to refine the latent factors of the two domains. In this way, we can achieve two benefits: (i) The respective intrinsic geometric property within two domains can be preserved, which allows the learning model to carefully follow the domain-specific data distribution and fundamentally alleviate the negative transfer problem. (ii) When the target data is extremely sparse, the neighborhood structure information between entities (users or items) is inaccurate, while the relatively dense auxiliary binary rating data has relatively more accurate neighborhood structure information. Transferring the neighborhood structure information of entities in the auxiliary data to the target data can help the target data obtain more accurate neighborhood structure information and further alleviate the under-transfer issue.

User Graph Regularization From geometric perspective, the data point is usually sampled from a low dimensional manifold embedded in a high-dimensional ambient space [4, 5]. By the manifold assumption [2], if two users $x_{i\cdot}^{\tau}$ and $x_{j\cdot}^{\tau}$ are close in the intrinsic geometry of the data distribution, then their embedding $u_{i\cdot}^{\tau}$ and $u_{j\cdot}^{\tau}$ are also close to each other. This geometric structure of scattered data can be effectively encoded by p-nearest neighbor graphs. Consider a user graph $\mathcal{G}_U^{\tau} = (\mathcal{V}_U^{\tau}, \mathcal{E}_U^{\tau})$, whose vertex set $\mathcal{V}_U^{\tau}$ corresponds to users $\{x_{1\cdot}^{\tau}, \cdots, x_{M\cdot}^{\tau}\}$. The symmetric adjacency matrix of $\mathcal{G}_U^{\tau}$ can be defined as

$$(W_U^{\tau})_{ij} = \begin{cases} sim(x_{i\cdot}^{\tau}, x_{j\cdot}^{\tau}), if\ x_{j\cdot}^{\tau} \in \mathcal{N}(x_{i\cdot}^{\tau})\ or\ x_{i\cdot}^{\tau} \in \mathcal{N}(x_{j\cdot}^{\tau}) \\ 0, \qquad otherwise, \end{cases} \tag{6}$$

where $\mathcal{N}(x_{i\cdot}^{\tau})$ denotes the k-nearest neighbor of $x_{i\cdot}^{\tau}$. $sim(\cdot,\cdot)$ is an appropriate similarity function for calculating similarities between users. sim can be cosine similarity, Pearson correlation coefficient or the radial basis function (RBF) measurement, etc. In our experiments, for the convenience of calculation, we used the following cosine similarity measurement: $sim(x_i^{\tau}, x_j^{\tau}) = \frac{x_i^{\tau} \cdot x_j^{\tau}}{\|x_i^{\tau}\| \|x_j^{\tau}\|}$. We use Euclidian distance to measure the closeness between each pair of embeddings $u_{i\cdot}^{\tau}$ and $u_{j\cdot}^{\tau}$, i.e.,$\|u_{i\cdot}^{\tau} - u_{j\cdot}^{\tau}\|_2^2$. According to Cai et al. [4], using the graph $\mathcal{G}_U^{\tau}$ to maintain the geometry structure in domain $\mathcal{D}_{\tau}$ can be accomplished via the user graph regularization as follows

$$\begin{aligned} \mathcal{R}(U_{\tau}) &= \frac{1}{2}\sum_{i,j} \left\| u_{i\cdot}^{\tau} - u_{j\cdot}^{\tau} \right\|_2^2 (W_U^{\tau})_{ij} = \sum_{i,j} u_{i\cdot}^{\tau} (W_U^{\tau})_{ij} u_{i\cdot}^{\tau T} - \sum_{i,j} u_{i\cdot}^{\tau} (W_U^{\tau})_{ij} u_{j\cdot}^{\tau T} \\ &= \sum_{i} u_{i\cdot}^{\tau} (D_U^{\tau})_{ii} u_{i\cdot}^{\tau T} - \sum_{i,j} u_{i\cdot}^{\tau} (W_U^{\tau})_{ij} u_{j\cdot}^{\tau T} = tr\left(U_{\tau}^T \left(D_U^{\tau} - W_U^{\tau} \right) U_{\tau} \right) \\ &= tr(U_{\tau}^T L_U^{\tau} U_{\tau}), \end{aligned} \tag{7}$$

where $D_U^{\tau} = diag(\sum_j (W_U^{\tau})_{ij})$ is a diagonal matrix, and $L_U^{\tau} = D_U^{\tau} - W_U^{\tau}$ is the graph Laplacian matrix of user graph.

Item Graph Regularization Similar to user graph regularization, by the manifold assumption [2], if two items $x_{\cdot i}^{\tau}$ and $x_{\cdot j}^{\tau}$ are close in the intrinsic geometry of the data distribution, then their embedding $v_{i\cdot}^{\tau}$ and $v_{j\cdot}^{\tau}$ are also close to each other. Consider an item graph

$\mathcal{G}_V^\tau = (\mathcal{V}_V^\tau, \mathcal{E}_V^\tau)$, whose vertex set $\mathcal{V}_V^\tau$ corresponds to items $\{x_{\cdot 1}^\tau, \cdots, x_{\cdot N}^\tau\}$. The symmetric adjacency matrix of $\mathcal{G}_V^\tau$ can be defined as

$$(W_V^\tau)_{ij} = \begin{cases} sim(x_{\cdot i}^\tau, x_{\cdot j}^\tau), if\ x_{\cdot j}^\tau \in \mathcal{N}(x_{\cdot i}^\tau)\ or\ x_{\cdot i}^\tau \in \mathcal{N}(x_{\cdot j}^\tau) \\ 0, \qquad\qquad\qquad\qquad otherwise. \end{cases} \tag{8}$$

According to Cai et al. [4] using the graph $\mathcal{G}_V^\tau$ to preserve the geometry structure in domain $\mathcal{D}_\tau$ can be achieved by the item graph regularization as follows

$$\begin{aligned} \mathcal{R}(V_\tau) &= \frac{1}{2}\sum_{i,j}\left\| v_{i\cdot}^\tau - v_{j\cdot}^\tau \right\|_2^2 (W_V^\tau)_{ij} = \sum_{i,j} v_{i\cdot}^\tau (W_V^\tau)_{ij} v_{i\cdot}^{\tau T} - \sum_{i,j} v_{i\cdot}^\tau (W_V^\tau)_{ij} v_{j\cdot}^{\tau T} \\ &= \sum_i v_{i\cdot}^\tau (D_V^\tau)_{ii} v_{i\cdot}^{\tau T} - \sum_{i,j} v_{i\cdot}^\tau (W_V^\tau)_{ij} v_{j\cdot}^{\tau T} = tr\left(V_\tau^T \left(D_V^\tau - W_V^\tau\right) V_\tau\right) \\ &= tr(V_\tau^T L_V^\tau V_\tau), \end{aligned} \tag{9}$$

where $D_V^\tau = diag(\sum\limits_j (W_V^\tau)_{ij})$ is a diagonal matrix, and $L_V^\tau = D_V^\tau - W_V^\tau$ is the graph Laplacian matrix of item graph.

We call the graph regularization terms in (7) and (9) the graph co-regularization of user and item graphs (GCRUI). On the one hand, GCRUI is to maintain the intrinsic geometric structure on users and items simultaneously for alleviating the negative transfer issue. On the other hand, when the target data is extremely sparse, relatively dense and accurate neighborhood structure information between entities in auxiliary data can be transferred to the target data, to further alleviate the under-transfer issue.

3.3 Optimization framework

In order to further improve the predictive performance of cross-domain recommendations, we should consider these two learning objectives together. The reasons are: (i) with weighted collective matrix tri-factorization, the common latent factors of users and items and the common partial cluster-level rating pattern are extracted, through which more knowledge can be transferred between domains. However, while enhancing the knowledge transfer, negative information of the auxiliary data may also be introduced into the target data, resulting in a negative transfer issue. (ii) with GCRUI, the intrinsic geometric structure of each domain can be preserved to alleviate negative transfer issue. In addition, when the target data is extremely sparse, the neighborhood structure information in the auxiliary data will be transferred to the target data to help it obtain more accurate neighborhood structure information, thereby further alleviating the under-transfer issue.

Based on the above considerations, we seamlessly integrate the weighted collective matrix tri-factorization and GCRUI into a unified framework, and obtain the following objective function

$$\begin{aligned} \min_{U_t, U_a, V_t, V_a, S, S_t, S_a \geq 0} \quad & \left\| Y_t \odot (X_t - U_t[S, S_t]V_t{}^T) \right\|_F^2 \\ & + \lambda \left\| Y_a \odot (X_a - U_a[S, S_a]V_a{}^T) \right\|_F^2 \\ & + \theta_u \left\| U_t - U_a \right\|_F^2 + \theta_v \left\| V_t - V_a \right\|_F^2 \\ & + \alpha_u tr(U_t^T L_U^t U_t) + \alpha_v tr(V_t^T L_V^t V_t) \\ & + \beta_u tr(U_a^T L_U^a U_a) + \beta_v tr(V_a^T L_V^a V_a), \end{aligned} \tag{10}$$

where α_u and α_v are the graph regularization parameters of the user graph and item graph from the target data, respectively. β_u and β_v are the graph regularization parameters of the user graph and item graph from the auxiliary data, respectively.

Finally, we impose the Frobenius norm on S, S_t and S_a to avoid overfitting. We end up with the following minimization problem for EKT

$$\begin{aligned}
\min_{U_t, U_a, V_t, V_a, S, S_t, S_a \geq 0} J = & \left\| Y_t \odot (X_t - U_t[S, S_t]{V_t}^T) \right\|_F^2 \\
& + \lambda \left\| Y_a \odot (X_a - U_a[S, S_a]{V_a}^T) \right\|_F^2 \\
& + \theta_u \left\| U_t - U_a \right\|_F^2 + \theta_v \left\| V_t - V_a \right\|_F^2 \\
& + \alpha_u tr(U_t^T L_U^t U_t) + \alpha_v tr(V_t^T L_V^t V_t) \\
& + \beta_u tr(U_a^T L_U^a U_a) + \beta_v tr(V_a^T L_V^a V_a) \\
& + \alpha_s \left\| S_t \right\|_F^2 + \beta_s \left\| S_a \right\|_F^2 + \gamma_s \left\| S \right\|_F^2 ,
\end{aligned} \tag{11}$$

where α_s, β_s and γ_s are the regularization parameters for controlling the strength of regularization.

3.4 Learning the EKT

In order to facilitate the optimization of the EKT method, we rewrite the (11) as

$$\begin{aligned}
\min_{U_t, U_a, V_t, V_a, S, S_t, S_a \geq 0} J = & \left\| Y_t \odot (X_t - U_t S V_{t_0}^T - U_t S_t V_{t_1}^T) \right\|_F^2 \\
& + \lambda \left\| Y_a \odot (X_a - U_t S V_{a_0}^T - U_t S_a V_{a_1}^T) \right\|_F^2 \\
& + \theta_u \left\| U_t - U_a \right\|_F^2 + \theta_v \left\| V_t - V_a \right\|_F^2 \\
& + \alpha_u tr(U_t^T L_U^t U_t) + \alpha_v tr(V_t^T L_V^t V_t) \\
& + \beta_u tr(U_a^T L_U^a U_a) + \beta_v tr(V_a^T L_V^a V_a) \\
& + \alpha_s \left\| S_t \right\|_F^2 + \beta_s \left\| S_a \right\|_F^2 + \gamma_s \left\| S \right\|_F^2 ,
\end{aligned} \tag{12}$$

where $U_t, U_a \in \mathbb{R}^{M \times d_1}$, $V_t = [V_{t_0}, V_{t_1}] \in \mathbb{R}^{N \times d_2}$, $V_a = [V_{a_0}, V_{a_1}] \in \mathbb{R}^{N \times d_2}$, $S \in \mathbb{R}^{d_1 \times c}$, $S_t \in \mathbb{R}^{d_1 \times (d_2 - c)}$, $S_a \in \mathbb{R}^{d_1 \times (d_2 - c)}$. The optimization of our proposed EKT method can be solved by an alternating minimization algorithm. Specifically, we optimize a variable and compute its update rule while fixing the remaining variables. The procedure is repeated until convergence.

Learning S, S_a and S_t Fix other variables to solve S, then we can rewrite the objective function in (12) as

$$\begin{aligned}
\min_{S \geq 0} J(S) = & \left\| Y_t \odot (X_t - U_t S V_{t_0} - U_t S_t V_{t_1}^T) \right\|_F^2 \\
& + \lambda \left\| Y_a \odot (X_a - U_a S V_{a_0} - U_a S_a V_{a_1}^T) \right\|_F^2 + \gamma_s \left\| S \right\|_F^2 .
\end{aligned}$$

The derivative of $J(S)$ in regard to S is as follows

$$\begin{aligned}
\frac{\partial J(S)}{\partial S} = & 2U_t^T (Y_t \odot (-X_t + U_t S V_{t_0}^T + U_t S_t V_{t_1}^T)) V_{t_0} \\
& + 2\lambda U_a^T (Y_a \odot (-X_a + U_a S V_{a_0}^T + U_a S_a V_{a_1}^T)) V_{a_0} + 2\gamma_s S.
\end{aligned}$$

Utilizing the Karush-Kuhn-Tucker(KKT) complementary condition for the nonnegativity of S and letting $\frac{\partial J(S)}{\partial S} = 0$, we can get

$$\begin{aligned}&[2U_t^T(Y_t \odot (-X_t + U_t S V_{t_0}^T + U_t S_t V_{t_1}^T))V_{t_0} + 2\gamma_s S\\ &+2\lambda U_a^T(Y_a \odot (-X_a + U_a S V_{a_0}^T + U_a S_a V_{a_1}^T))V_{a_0}] \odot S = 0.\end{aligned}$$

Then we can obtain the updating rule for S as follows

$$S \leftarrow S \odot \frac{\left[U_t^T\left(Y_t \odot X_t\right) V_{t_0} + \lambda U_a^T(Y_a \odot X_a)V_{a_0}\right]}{[A+B]} \tag{13}$$

$$\begin{aligned}A &= U_t^T(Y_t \odot (U_t S V_{t_0}^T + U_t S_t V_{t_1}^T))V_{t_0}\\ B &= \lambda U_a^T(Y_a \odot (U_a S V_{a_0}^T + U_a S_a V_{a_1}^T))V_{a_0} + \gamma_s S,\end{aligned}$$

where $\frac{[\cdot]}{[\cdot]}$ denotes element-wise division. Similarly, we can obtain the following updating rules for learning S_t and S_a

$$S_t \leftarrow S_t \odot \frac{[U_t^T\left(Y_t \odot X_t\right) V_{t_1}]}{[U_t^T((Y_t \odot (U_t S V_{t_0}^T + U_t S_t V_{t_1}^T))V_{t_1} + \alpha_s S_t]} \tag{14}$$

$$S_a \leftarrow S_a \odot \frac{[\lambda U_a^T(Y_a \odot X_a)V_{a_1}]}{[\lambda U_a^T(Y_a \odot (U_a S V_{a_0}^T + U S_a V_{a_1}^T))V_{a_1} + \beta_s S_a]}. \tag{15}$$

Learning U_t, U_a, V_t and V_a Similarly, fix other variables to solve U_t, then we rewrite the objective function in (11) as

$$\begin{aligned}\min_{U_t \geq 0} J\left(U_t\right) = &\left\| Y_t \odot (X_t - U_t[S, S_t]V_t{}^T)\right\|_F^2\\ &+ \theta_u \left\|U_t - U_a\right\|_F^2 + \alpha_u tr(U_t{}^T L_U^t U_t).\end{aligned}$$

The derivative of $J(U_t)$ with respect to U_t is as follows

$$\frac{\partial J\left(U_t\right)}{\partial U_t} = 2(Y_t \odot (-X_t + U_t[S, S_t]V_t^T))V_t[S, S_t]^T + 2\theta_u(U_t - U_a) + 2\alpha_u L_U^t U_t.$$

Using the Karush-Kuhn-Tucker(KKT) complementary condition for the nonnegativity of U_t and letting $\frac{\partial J(U_t)}{\partial U_t} = 0$, we can get

$$[(Y_t \odot (-X_t + U_t[S, S_t]V_t^T))V_t[S, S_t]^T + \theta_u(U_t - U_a) + \alpha_u L_U^t U_t] \odot U_t = 0.$$

Since L_U^t may take any signs, we decompose it as L_U^t = L_U^{t+} - L_U^{t-}, where $L_U^{t+} = \frac{1}{2}(|L_U^t| + L_U^t)$, L_U^{t-} = $\frac{1}{2}\left(\left|L_U^t\right|\text{ - }L_U^t\right)$, then

$$\begin{aligned}&[(Y_t \odot (-X_t + U_t[S, S_t]V_t^T))V_t[S, S_t]^T + \theta_u(U_t - U_a) + \alpha_u L_U^{t+} U_t\\ &-\alpha_u L_U^{t-} U_t] \odot U_t = 0.\end{aligned}$$

We obtain the updating rule for learning U_t as follows

$$U_t \leftarrow U_t \odot \frac{[(Y_t \odot X_t)\, V_t[S, S_t]^T + \theta_u U_a + \alpha_u L_U^{t-} U_t]}{[\left(Y_t \odot \left(U_t[S, S_t]V_t^T\right)\right) V_t[S, S_t]^T + \theta_u U_t + \alpha_u L_U^{t+} U_t]}. \tag{16}$$

The latent factor U_a, V_t and V_a can be learned in the similar way as for constrained optimization. We can get the updating rule for learning U_a, V_t and V_a as follows:

$$U_a \leftarrow U_a \odot \frac{[\lambda\,(Y_a \odot X_a)\, V_a[S, S_a]^T + \theta_u U_t + \beta_u L_U^{a-} U_a]}{[\lambda\left(Y_a \odot \left(U_a[S, S_a]V_a^T\right)\right) V_a[S, S_a]^T + \theta_u U_a + \beta_u L_U^{a+} U_a]} \tag{17}$$

$$V_t \leftarrow V_t \odot \frac{\left[(Y_t \odot X_t)^T U_t \left[S, S_t\right] + \theta_v V_a + \alpha_v L_V^{t-} V_t\right]}{\left[\left(Y_t \odot \left(U_t \left[S, S_t\right] V_t^T\right)\right)^T U_t \left[S, S_t\right] + \theta_v V_t + \alpha_v L_V^{t+} V_t\right]} \tag{18}$$

$$V_a \leftarrow V_a \odot \frac{\left[(Y_a \odot X_a)^T U_a \left[S, S_a\right] + \theta_v V_t + \beta_v L_V^{a-} V_a\right]}{\left[\left(Y_a \odot \left(U_a \left[S, S_a\right] V_a^T\right)\right)^T U_a \left[S, S_a\right] + \theta_v V_a + \beta_v L_V^{a+} V_a\right]}, \tag{19}$$

where $L_U^a = L_U^{a+} - L_U^{a-}$, $L_U^{a+} = \frac{1}{2}(|L_U^a| + L_U^a)$, $L_U^{a-} = \frac{1}{2}(|L_U^a| - L_U^a)$, $L_V^t = L_V^{t+} - L_V^{t-}$, $L_V^{t+} = \frac{1}{2}(|L_V^t| + L_V^t)$, $L_V^{t-} = \frac{1}{2}(|L_V^t| - L_V^t)$, $L_V^a = L_V^{a+} - L_V^{a-}$, $L_V^{a+} = \frac{1}{2}(|L_V^a| + L_V^a)$, $L_V^{a-} = \frac{1}{2}(|L_V^a| - L_V^a)$. Note that $V_{t_0} = V_t(:, 1:c)$, $V_{t_1} = V_t(:, c+1:d_2)$, $V_{a_0} = V_a(:, 1:c)$, $V_{a_1} = V_a(:, c+1:d_2)$.

Theorem 1 *Updating S, S_t, S_a, U_t, U_a, V_t and V_a sequentially by* (13)∼ (19) *will monotonically decrease the objective function in* (11) *until convergence.*

We will prove Theorem 1 in Section 3.5. The learning algorithm for EKT is summarized in Algorithm 1.

Algorithm 1 EKT:Enhanced knowledge transfer for collaborative filtering with multi-source heterogeneous feedbacks.

```
Input:
    $X_t$, the numerical rating matrix in target domain;
    $X_a$, the binary rating matrix in auxiliary domain;
    $Y_t$, the indicator matrix in target domain;
    $Y_a$, the indicator matrix in auxiliary domain;
    $W_U^t$, the user similarity graph in target domain;
    $W_V^t$, the item similarity graph in target domain;
    $W_U^a$, the user similarity graph in auxiliary domain;
    $W_V^a$, the item similarity graph in auxiliary domain;
Output:
    $U_t$, the user feature matrix in target domain;
    $V_t$, the item feature matrix in target domain;
    $U_a$, the user feature matrix in auxiliary domain;
    $V_a$, the item feature matrix in auxiliary domain;
    $S$, the shared user-item rating pattern matrix;
    $S_t$, the target data-specific rating pattern matrix;
    $S_a$, the auxiliary data-specific rating pattern matrix;
Step 1. Scale ratings in $X_t((X_t)_{ui} = ((X_t)_{ui}-0.5)/4.5$ or $(X_t)_{ui} = ((X_t)_{ui}-1)/4$, $(Y_t)_{ui} = 1, u = 1, \cdots, M, i = 1, \cdots, N)$.
Step 2. Randomly initialize $U_t$, $V_t$, $U_a$, $V_a$, $S$, $S_t$, $S_a$.
Step 3. Update $U_t$, $V_t$, $U_a$, $V_a$, $S$, $S_t$, $S_a$.
for iter 1 to $K$ do
    Step 3.1. Fix $U_t$, $V_t$, $U_a$, $V_a$, $S_t$ and $S_a$, update $S$ as show in (13).
    Step 3.2. Fix $U_t$, $V_t$, $U_a$, $V_a$, $S$ and $S_a$, update $S_t$ as show in (14).
    Step 3.3. Fix $U_t$, $V_t$, $U_a$, $V_a$, $S$ and $S_t$, update $S_a$ as show in (15).
    Step 3.4. Fix $V_t$, $U_a$, $V_a$, $S$, $S_t$ and $S_a$, update $U_t$ as show in (16).
    Step 3.5. Fix $U_t$, $V_t$, $V_a$, $S$, $S_t$ and $S_a$, update $U_a$ as show in (17).
    Step 3.6. Fix $U_t$, $U_a$, $V_a$, $S$, $S_t$ and $S_a$, update $V_t$ as show in (18).
    Step 3.7. Fix $U_t$, $U_a$, $V_t$, $S$, $S_t$ and $S_a$, update $V_a$ as show in (19).
end for
```

3.5 Convergence analysis

We can adopt the auxiliary function approach [4, 16, 21] to prove Theorem 1. For simplicity, we will only prove that the objective function J in (11) decreases monotonically under the update rule for U_t in (16). We can prove the convergence of the other update rules in a similar way. The definition of auxiliary function is described as follows

Definition 1 [16] $Q(p, p')$ is an auxiliary function for $F(p)$ if the conditions

$$Q(p, p') \geq F(p) \quad and \quad Q(p, p) = F(p)$$

are satisfied for any given p, p' .

Lemma 1 [16] *If Q is an auxiliary function for F, then F is decreasing under the update*

$$p^{(k+1)} = \arg\min_{p} Q\left(p, p^{(k)}\right). \tag{20}$$

Proof

$$F(p^{(k+1)}) \leq Q(p^{(k+1)}, p^{(k)}) \leq Q(p^{(k)}, p^{(k)}) = F(p^{(k)}).$$

Next, by constructing an appropriate auxiliary function, we will demonstrate that (16) is exactly the update rule of Lemma 1. For any element u_{ij} in U_t, F_{ij} is used to represent the part of J that is only relevant to u_{ij}. We compute the corresponding first and second derivatives of F_{ij} in regard to u_{ij} as follows □

$$\begin{aligned} F'_{ij} &= \Big[2\left(Y_t \odot \left(-X_t + U_t[S, S_t]V_t{}^T\right)\right) V_t[S, S_t]^T \\ &\quad + 2\theta_u(U_t - U_a) + 2\alpha_u L_U^{t+} U_t - 2\alpha_u L_U^{t-} U_t\Big]_{ij} \end{aligned}$$

$$F''_{ij} = 2(Y_t)_{ij}\left(\left([S, S_t]V_t{}^T\right) V_t[S, S_t]^T\right)_{jj} + 2\theta_u + 2\alpha_u\left(L_U^{t+}\right)_{ii} - 2\alpha_u\left(L_U^{t-}\right)_{ii}.$$

Lemma 2 *Function*

$$\begin{aligned} Q(u, u_{ij}^{(k)}) &= F_{ij}(u_{ij}^{(k)}) + F'_{ij}(u_{ij}^{(k)})(u - u_{ij}^{(k)}) \\ &\quad + \frac{\left(2\left(Y_t \odot \left(U_t[S, S_t]V_t{}^T\right)\right) V_t[S, S_t]^T + 2\theta_u U_t + 2\alpha_u L_U^{t+} U_t\right)_{ij}}{u_{ij}^{(k)}} \cdot (u - u_{ij}^{(k)})^2 \end{aligned} \tag{21}$$

is an appropriate auxiliary function for $F_{ij}(u)$.

Proof It is straightforward that $Q(u, u) = F_{ij}(u)$, and hence we only need verify that $Q\left(u, u_{ij}^{(k)}\right) \geq F_{ij}(u)$. To achieve this, we use Taylor series to expand $F_{ij}(u)$

$$\begin{aligned} F_{ij}(u) &= F_{ij}\left(u_{ij}^{(k)}\right) + F'_{ij}\left(u_{ij}^{(k)}\right)\left(u - u_{ij}^{(k)}\right) \\ &\quad + \left(2(Y_t)_{ij}\left(\left([S, S_t]V_t{}^T\right) V_t[S, S_t]^T\right)_{jj} + 2\theta_u \right. \\ &\quad \left. + 2\alpha_u\left(L_U^{t+}\right)_{ii} - 2\alpha_u\left(L_U^{t-}\right)_{ii}\right) \cdot \left(u - u_{ij}^{(k)}\right)^2. \end{aligned}$$

Through algebra operations, we can get two inequalities:

$$\begin{aligned} F_{ij}\left(u\right) =\; & F_{ij}\left(u_{ij}^{(k)}\right) + F_{ij}'\left(u_{ij}^{(k)}\right)\left(u - u_{ij}^{(k)}\right) \\ & + \left(2(Y_t)_{ij}\left(\left([S, S_t]V_t{}^T\right) V_t[S, S_t]^T\right)_{jj} + 2\theta_u \right. \\ & \left. + \; 2\alpha_u\left(L_U^{t+}\right)_{ii} \; -2\alpha_u\left(L_U^{t-}\right)_{ii}\right) \cdot \left(u - u_{ij}^{(k)}\right)^2. \end{aligned}$$

$$\left(L_U^{t+}U_t\right)_{ij} = \sum_q u_{qj}^{(k)}\left(L_U^{t+}\right)_{iq} \geq u_{ij}^{(k)}\left(L_U^{t+}\right)_{ii}.$$

By jointly comparing the above two inequalities, we have

$$\begin{aligned} & \frac{\left(2\left(Y_t \odot \left(U_t[S, S_t]V_t^T\right)\right) V_t[S, S_t]^T + 2\theta_u U_t + 2\alpha_u L_U^{t+} U_t\right)_{ij}}{u_{ij}^{(k)}} \\ & \geq 2(Y_t)_{ij}\left(\left([S, S_t]\, V_t^T\right) V_t[S, S_t]^T\right)_{ij} + 2\theta_u + 2\alpha_u\left(L_U^{t+}\right)_{ii} - 2\alpha_u\left(L_U^{t-}\right)_{ii}. \end{aligned}$$

Further, we can get $Q\left(u, u_{ij}^{(k)}\right) \geq F_{ij}\left(u\right)$, and Lemma 2 holds. □

Proof of Theorem 1 According to Lemmas 1 and 2, we can obtain the update rule for U_t by minimizing $Q\left(u_{ij}^{(k+1)}, u_{ij}^{(k)}\right)$. Setting $\frac{\partial Q\left(u_{ij}^{(k+1)}, u_{ij}^{(k)}\right)}{\partial u_{ij}^{(k+1)}} = 0$, we get

$$\begin{aligned} u_{ij}^{(k+1)} &= u_{ij}^{(k)} - \frac{u_{ij}^{(k)} F_{ij}'\left(u_{ij}^{(k)}\right)}{\left(2\left(Y_t \odot \left(U_t[S, S_t]V_t^T\right)\right) V_t[S, S_t]^T + 2\theta_u U_t + 2\alpha_u L_U^{t+} U_t\right)_{ij}} \\ &= u_{ij}^{(k)} \frac{\left(\left(Y_t \odot X_t\right) V_t[S, S_t]^T + 2\theta_u U_a + \alpha_u L_U^{t-} U_t\right)_{ij}}{\left(\left(Y_t \odot \left(U_t[S, S_t]V_t{}^T\right)\right) V_t[S, S_t]^T + 2\theta_u U_t + \alpha_u L_U^{t+} U_t\right)_{ij}}. \end{aligned}$$

This above update rule is uniform with (16). For each iteration of updating, we can obtain

$$\begin{aligned} J\left(U_t^{(0)}\right) =\; & Q\left(U_t^{(0)}, U_t^{(0)}\right) \geq Q\left(U_t^{(1)}, U_t^{(0)}\right) \\ \geq\; & Q\left(U_t^{(1)}, U_t^{(1)}\right) = J\left(U_t^{(1)}\right) \geq \cdots \geq J\left(U_t^{(T)}\right). \end{aligned}$$

Therefore, $J(U)$ is monotonically decreasing during iterations. Since the objective function in (11) is obviously bounded below, we prove the convergence of Theorem 1. □

3.6 Analysis

When $\theta_u = \theta_v = 0$, $\alpha_u = \alpha_v = 0$, $\beta_u = \beta_v = 0$, EKT reduces to CLFM [10], which does not involve the similarity constraints of latent factor and the co-graph regularization of user and item graphs. When $c = 0$, $\theta_u = \theta_v = 0$, $\beta_u = \beta_v = 0$, EKT reduces to GWNMTF [11], which focuses on learning the latent factors of users and items and user-item rating pattern using only target rating data. When $c = 0$, $\theta_u = \theta_v = 0$, $\beta_u = \beta_v = 0$, $\alpha_u = \alpha_v = 0$, EKT reduces to WNMTF, which does not involve the user and item graphs. Therefore, our EKT is generic and absorbs CLFM, GWNMTF and WNMTF as special cases.

Table 2 Description of subset of ML10M data (M=N=5000) and subset of Netflix data (M=N=5000)

Date set		Form	Sparsity
ML10M	target(training)	$\{0.5, 1, 1.5, \cdots, 5, ?\}$	$\leq 1\%$
(subset)	target(test)	$\{0.5, 1, 1.5, \cdots, 5, ?\}$	7.04%
	auxiliary	$\{0, 1, ?\}$	2%
Netflix	target(training)	$\{1, 2, 3, 4, 5, ?\}$	$\leq 1\%$
(subset)	target(test)	$\{1, 2, 3, 4, 5, ?\}$	16.2%
	auxiliary	$\{0, 1, ?\}$	2%

4 Experiments

4.1 Data sets and evaluation metrics

4.1.1 Data sets

We evaluate our proposed method using the subset of two movie rating data sets MovieLens10M[1] (denoted as ML10M) and Netflix[2]. The ML10M rating data contains more than 10^7 rating with values in $\{0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5\}$, which are given by more 7.1×10^4 users on around 1.1×10^5 movies. Similar to the experimental methods of Pan et.al. [31], we randomly extract a 5000×5000 dense rating matrix R from ML10M data, and then R is randomly split into training set R_T and test set R_E each with a 50% ratings. R_E remains unchanged, while target training data X_t with different sparse ratings of 0.01%, 0.05%, 0.1%, 0.5% and 1% are constructed by randomly sampling corresponding numbers of observed rating of 2500, 12500, 25000, 125000 and 250000 from R_T. The auxiliary data X_a is constructed by randomly sampling 100 observed ratings on average from R_T for each user. A pre-processing approach[34] is adopted by relabeling ratings with value less than 4 in X_a as 0 (dislike), and then ratings with value greater than or equal to 4 as 1 (like). The overlap between X_t and X_a are 0.0015%, 0.0075%, 0.015%, 0.075% and 0.15% correspondingly.

The Netflix rating data contains more than 10^8 ratings with values in $\{1, 2, 3, 4, 5\}$, which are given by more than 4.8×10^5 users on around 1.8×10^4 movies. The data set employed in the experiments is constructed in the same way as the ML10M data. The final data sets are summed up in Table 2.

4.1.2 Evaluation metrics

We employed the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) as the evaluation metrics

$$MAE = \sum_{(u,i,x_{ui}) \in R_E} \left| x_{ui} - \hat{x}_{ui} \right| / |R_E| \tag{22}$$

$$RMSE = \sqrt{\sum_{(u,i,x_{ui}) \in R_E} \left(x_{ui} - \hat{x}_{ui} \right)^2 / |R_E|}, \tag{23}$$

[1] http://grouplens.org/datasets/movielens/

[2] http://www.netflix.com.

where $\hat{x}_{ui}$ and x_{ui} represent the predicted and true ratings, respectively, and $|R_E|$ represents the number of test ratings. When the required number of observed ratings are generated from R_T, we run 5 randomised trials, and report the average results.

4.2 Baselines and parameter settings

4.2.1 Baselines

In our experiments, we compare our proposed EKT with the following seven closely related baseline algorithms:

- WNMTF (Weighted Nonnegative Matrix Tri-Factorization) is a single-domain recommendation method that decomposes the target user-item rating matrix into the product of three low-rank non-negative matrices, which are then used to predict the rating and provide a recommendation.
- GWNMTF (Graph Regularized Weighted Nonnegative Matrix Tri-Factorization) [11] is a non-transfer learning method based on graph regularization. In this model, two graphs are constructed on user side and item side to utilize the internal information and external information.
- PMF (Probabilistic Matrix Factorization) [24] adopts a probabilistic model with Gaussian observation noise to learn the latent features of users and items by maximizing the conditional distribution of the latent feature matrix of users and items over the observed target ratings. It learns on the target domain only.
- TCF (Transfer by Collective Factorization) [30] is based on transfer learning method using binary auxiliary data. In this model, the shared latent space U and V are constructed in a collective manner, and the data-dependent effect is captured via learning inner matrices B, $\tilde{B}$ separately. In TCF, there are two variants namely CMTF (Collective Matrix Tri-Factorization) and CSVD (Collective SVD).
- iTCF (Interaction-rich Transfer by Collective Factorization) [28] is an efficient transfer learning algorithm in collaborative filtering with heterogeneous user feedbacks. In iTCF, Richer interactions are introduced by sharing both item-specific latent features and the predictability in two heterogeneous data in a smooth manner.
- TMF (Transfer by Mixed Factorization) [29] is a generic mixed factorization based transfer learning framework for collaborative recommendation with heterogeneous explicit feedbacks. TMF unifies two transfer methods in one optimization framework: instance-based transfer by intergrative factorization and feature-based transfer by collective factorization.
- WNMTF-TL (Weight Nonnegative Matrix Tri-Factorization based on Transfer Learning). Active Transfer Learning for Cross-System Recommendation [46] method is proposed to construct cross-domain entity correspondences and then the actively constructed entity correspondences plugged into a general matrix factorization model. In our problem formulate, cross-domain entities are one-to-one correspondence, therefore we removed the active learning module in the originally proposed method. In addition, for a fairer comparison with our EKT method, we adopt WNMTF as the matrix factorization model, and then use the entity similarity learned from the auxiliary binary rating data as a prior to constrain the entity similarity in the target domain. We name the method as WNMTF-TL.

Our EKT (Enhanced Knowledge Transfer for Collaborative Filtering with Multi-Source Heterogeneous Feedbacks) method seamlessly integrates the weighted collective matrix tri-factorization and graph co-regularization of user and item graphs into a unified framework that can simultaneously enhance knowledge transfer and alleviate negative transfer.

4.2.2 Parameter settings

For all methods, different numbers of latent factors $d_1, d_2 \in \{5, 10, 15, 20\}$ are tried. For EKT, different number of shared latent user-item rating patterns $c \in \{0, 1, \cdots, d_2\}$ are tried. Note that when $c = 0$, there is no shared user-item rating pattern between the auxiliary data and the target data, and when $c = d_2$, the latent user-item rating pattern is fully shared between the auxiliary data and the target data. For GWNMTF, different trade-off parameters $\lambda = \mu \in \{0.01, 0.1, 1, 10, 100\}$ are tried. For PMF, different trad-off parameters $\lambda_U = \lambda_V \in \{0.01, 0.1, 1\}$ are tried. For TCF (CMTM), β is fixed as 1, and different tradeoff parameters $\alpha_u = \alpha_v \in \{0.01, 0.1, 1\}$, $\lambda \in \{0.01, 0.1, 1\}$ are tried. For TCF (CSVD), different trade-off parameters $\lambda \in \{0.01, 0.1, 1\}$ are tried. For iTCF, we fixed the trade-off parameter $\lambda = 1$, $\rho = 0.5$, and different trade-off parameters $\alpha_u = \alpha_v \in \{0.01, 0.1, 1\}$, $\beta_u = \beta_v \in \{0.01, 0.1, 1\}$ are tried. For TMF, we fixed the trade-off parameter $\lambda = 1$, $\rho = 0.5$, $\delta_P = \delta_N = 1$, $w_p = 2$, $w_N = 1$, and different trade-off parameters $\alpha_u = \alpha_v \in \{0.01, 0.1, 1\}$, $\beta_u = \beta_v \in \{0.01, 0.1, 1, 10, 100\}$ are tried. For WNMTF-TL, different trade-off parameters $\lambda_C \in \{0.01, 0.1, 1, 10, 100\}$ are tried. For EKT, λ, α_s, β_s and γ_s are fixed as 1, different trade-off parameters of $\alpha_u = \alpha_v \in \{0.01, 0.1, 1\}$, $\beta_u = \beta_v \in \{0.01, 0.1, 1\}$, $\theta_u = \theta_v \in \{0.1, 0.5, 1, 5, 10\}$ are tried. Further analysis of the parameters is provided in Section 4.5. Note that for EKT, to alleviate the heterogeneity between the target data and the auxiliary data, the target rating matrices from ML10M or Neflix for training are preprocessed by letting $(X_t)_{ui} = ((X_t)_{ui} - 0.5)/4.5$ or $(X_t)_{ui} = ((X_t)_{ui} - 1)/4$, respectively. For iTCF and TMF, follow Pan et al. [28, 29], "0 (dislike)" and "1 (like)" in auxiliary binary rating data are replaced with numerical values of "1 (dislike)" and "5 (like)", respectively. For all baseline methods and our EKT method, each of them is run 1000 iterations, and the best results are reported.

4.3 Experimental results

The experimental results on ML10M and Netflix are shown in Tables 3 and 4, respectively. From these results, we can make the following observations:

(1) For the non-transfer learning methods, we can see that GWNMTF consistently outperforms WNMTF at all sparsity levels. In addition, GWNMTF is better than PMF when the sparsity is higher (e.g. $\geq$ 0.5% for ML10M and $\geq$ 0.1% for Netflix). However, when the sparsity is lower (e.g. $\leq$ 0.1% for ML10M and $\leq$ 0.05% for Netflix), the prediction performance of GWNMTF is worse than PMF. The reason is that when the target rating matrix is denser, the neighborhood structure information between entities (users or items) obtained is more accurate. Conversely, when the sparsity is lower, the neighborhood structure information between entities obtained may be inaccurate.

(2) Transfer learning technology is an effective method to solve the sparsity issue in collaborative filtering.

   (a) We can see that Transfer-based methods consistently outperform the non-transfer method at all sparsity levels. Prove the effectiveness of TL-based CF methods.

   (b) The proposed transfer learning methods of EKT performs better than all baseline methods at almost all sparsity levels except the denser 1% case, and the average

Table 3 Prediction performance comparison on the subset of MovieLens10M data. Numbers in boldface is the best result among all methods

Metrics	Methods	Sparsity					Average
		0.01%	0.05%	0.1%	0.5%	1%	
MAE	WNMTF	2.4272 ± 0.0089	1.2426 ± 0.0072	0.8871 ± 0.0040	0.6739 ± 0.0004	0.6575 ± 0.0003	1.1777 ± 0.0042
	GWNMTF	2.4118 ± 0.0103	1.2086 ± 0.0061	0.8245 ± 0.0038	0.6662 ± 0.0022	0.6395 ± 0.0002	1.1501 ± 0.0045
	PMF	1.0584 ± 0.0090	0.8738 ± 0.0019	0.7931 ± 0.0017	0.6742 ± 0.0004	0.6586 ± 0.0011	0.8116 ± 0.0028
	TCF(CMTF)	0.8648 ± 0.0016	0.8234 ± 0.0025	0.7559 ± 0.0021	0.6555 ± 0.0004	0.6350 ± 0.0003	0.7469 ± 0.0014
	TCF(CSVD)	0.9252 ± 0.0051	0.8311 ± 0.0029	0.7657 ± 0.0020	0.6524 ± 0.0002	0.6313 ± 0.0001	0.7611 ± 0.0021
	iTCF	0.8325 ± 0.0018	0.7793 ± 0.0013	0.7411 ± 0.0024	0.6549 ± 0.0009	0.6370 ± 0.0012	0.7290 ± 0.0015
	TMF	0.7543 ± 0.0039	0.7232 ± 0.0011	0.7041 ± 0.0015	0.6505 ± 0.0003	**0.6262 ± 0.0003**	0.6916 ± 0.0014
	WNMTF-TL	0.8118 ± 0.0015	0.7526 ± 0.0007	0.7186 ± 0.0014	0.6540 ± 0.0005	0.6354 ± 0.0004	0.7145 ± 0.0009
	EKT	**0.6754 ± 0.0016**	**0.6727 ± 0.0037**	**0.6713 ± 0.0007**	**0.6448 ± 0.0003**	0.6275 ± 0.0005	**0.6583 ± 0.0014**
RMSE	WNMTF	2.7318 ± 0.0007	1.6499 ± 0.0079	1.1899 ± 0.0054	0.8714 ± 0.0007	0.8494 ± 0.0003	1.4585 ± 0.0030
	GWNMTF	2.7198 ± 0.0079	1.6211 ± 0.0075	1.1166 ± 0.0057	0.8643 ± 0.0023	0.8310 ± 0.0003	1.4306 ± 0.0047
	PMF	1.2899 ± 0.0113	1.1133 ± 0.0035	1.0167 ± 0.0025	0.8675 ± 0.0006	0.8471 ± 0.0008	1.0269 ± 0.0037
	TCF(CMTF)	1.0618 ± 0.0023	1.0536 ± 0.0041	0.9727 ± 0.0022	0.8470 ± 0.0007	0.8242 ± 0.0005	0.9519 ± 0.0020
	TCF(CSVD)	1.1678 ± 0.0070	1.0566 ± 0.0044	0.9778 ± 0.0032	0.8431 ± 0.0005	0.8182 ± 0.0002	0.9727 ± 0.0034
	iTCF	1.0431 ± 0.0021	0.9935 ± 0.0016	0.9481 ± 0.0026	0.8457 ± 0.0006	0.8235 ± 0.0004	0.9308 ± 0.0015
	TMF	0.9606 ± 0.0048	0.9278 ± 0.0017	0.9058 ± 0.0021	0.8428 ± 0.0004	**0.8130 ± 0.0005**	0.8900 ± 0.0019
	WNMTF-TL	1.0261 ± 0.0017	0.9598 ± 0.0007	0.9210 ± 0.0014	0.8480 ± 0.0006	0.8254 ± 0.0003	0.9161 ± 0.0009
	EKT	**0.8718 ± 0.0005**	**0.8675 ± 0.0007**	**0.8645 ± 0.0004**	**0.8334 ± 0.0004**	0.8148 ± 0.0005	**0.8504 ± 0.0005**

Table 4 Prediction performance comparison on the subset of Netflix data. Numbers in boldface is the best result among all methods

Metrics	Methods	Sparsity					Average
		0.01%	0.05%	0.1%	0.5%	1%	
MAE	WNMTF	2.1492 ± 0.0020	1.2203 ± 0.0054	0.9180 ± 0.0029	0.7368 ± 0.0003	0.7219 ± 0.0004	1.1480 ± 0.0022
	GWNMTF	2.1371 ± 0.0033	1.1295 ± 0.0066	0.8597 ± 0.0038	0.7314 ± 0.0016	0.6989 ± 0.0005	1.1113 ± 0.0032
	PMF	1.1266 ± 0.0169	0.9777 ± 0.0041	0.8724 ± 0.0018	0.7375 ± 0.0007	0.7202 ± 0.0020	0.8869 ± 0.0051
	TCF(CMTF)	0.8489 ± 0.0061	0.7939 ± 0.0011	0.7673 ± 0.0021	0.7057 ± 0.0008	0.6911 ± 0.0007	0.7614 ± 0.0022
	TCF(CSVD)	0.8334 ± 0.0036	0.7846 ± 0.0011	0.7565 ± 0.0017	0.6990 ± 0.0003	**0.6843 ± 0.0007**	0.7516 ± 0.0015
	iTCF	0.8839 ± 0.0020	0.8424 ± 0.0013	0.8005 ± 0.0011	0.7158 ± 0.0008	0.6959 ± 0.0005	0.7877 ± 0.0011
	TMF	0.8159 ± 0.0044	0.7628 ± 0.0012	0.7494 ± 0.0004	0.7036 ± 0.0003	0.6855 ± 0.0008	0.7434 ± 0.0014
	WNMTF-TL	0.8877 ± 0.0009	0.8294 ± 0.0014	0.7974 ± 0.0006	0.7229 ± 0.0003	0.6954 ± 0.0006	0.7866 ± 0.0008
	EKT	**0.7335 ± 0.0014**	**0.7320 ± 0.0003**	**0.7287 ± 0.0005**	**0.6969 ± 0.0004**	0.6855 ± 0.0004	**0.7153 ± 0.0006**
RMSE	WNMTF	2.4437 ± 0.0017	1.5735 ± 0.0070	1.1939 ± 0.0044	0.9399 ± 0.0004	0.9199 ± 0.0005	1.4142 ± 0.0028
	GWNMTF	2.4344 ± 0.0026	1.4772 ± 0.0089	1.1125 ± 0.0061	0.9331 ± 0.0021	0.8946 ± 0.0006	1.3704 ± 0.0041
	PMF	1.3706 ± 0.0198	1.2363 ± 0.0055	1.1093 ± 0.0014	0.9383 ± 0.0002	0.9158 ± 0.0025	1.1141 ± 0.0059
	TCF(CMTF)	1.0635 ± 0.0065	1.0052 ± 0.0018	0.9735 ± 0.0030	0.9004 ± 0.0010	0.8824 ± 0.0008	0.9650 ± 0.0026
	TCF(CSVD)	1.0688 ± 0.0049	1.0055 ± 0.0016	0.9701 ± 0.0021	0.8964 ± 0.0005	**0.8766 ± 0.0007**	0.9634 ± 0.0020
	iTCF	1.0726 ± 0.0033	1.0428 ± 0.0013	1.0069 ± 0.0013	0.9104 ± 0.0005	0.8848 ± 0.0004	0.9835 ± 0.0014
	TMF	1.0289 ± 0.0044	0.9640 ± 0.0008	0.9499 ± 0.0005	0.9003 ± 0.0002	0.8773 ± 0.0010	0.9441 ± 0.0014
	WNMTF-TL	1.0735 ± 0.0045	1.0243 ± 0.0010	0.9940 ± 0.0012	0.9228 ± 0.0002	0.8900 ± 0.0005	0.9809 ± 0.0015
	EKT	**0.9300 ± 0.0005**	**0.9278 ± 0.0003**	**0.9249 ± 0.0003**	**0.8917 ± 0.0004**	0.8771 ± 0.0004	**0.9103 ± 0.0005**

prediction performance of our EKT method is significantly better than all baseline methods. In addition, it is interesting to see that, the sparser the target data is, our EKT method has more performance improvement than the baseline methods. The reason is that when the target data is extremely sparse, other baseline methods based on transfer learning are likely to encounter under-transfer and negative transfer issues, resulting in rapid degradation of performance. Our EKT method considers these two issues simultaneously in a unified framework, which not only can transfer more knowledge from the auxiliary data, but also effectively alleviate the negative transfer issue.

(c) CSVD performs better than CMTF at all sparsity levels of Netflix, which is consistent with the results reported in [31]. In addition, when the target data sparsity level is 1%, CSVD achieves slightly better prediction performance than our EKT methods, indicating that when the target data is not very sparse, the orthogonal constraint in CSVD can reduce noise, thereby improving the prediction performance. However, on ML10M, when the sparsity level is less than or equal to 0.1%, CSVD performs worse than the CMTF, which shows that when the sparsity level is lower, CSVD may be unstable and positive transfer may not be guaranteed.

(d) TMF performs better than iTCF at all sparsity levels, which is consistent with the results reported in [29]. In addition, at 1% sparse level of ML10M, TMF achieve slightly better predictive performance than our EKT method. The reason is that TMF utilizes the virtual user profile from the target data by combining the user liked items latent features and disliked items latent features. Therefore, when the target data is relatively denser, TMF can perform better.

(e) TMF performs better than CSVD at almost all sparsity levels except the denser case of 1% and 0.5% on Netflix. TMF performs better than CSVD when the sparsity is lower, because TMF introduces the global average preference scores, the user preference biases and the item preference biases into the prediction rules, which may help for an extremely sparse rating matrix.

(f) WNMTF-TL is significantly better than non-transfer learning methods of WNMTF and GWNMTF, which shows that transferring the similarity between entities estimated in the auxiliary data to the target data can improve the prediction performance of WNMTF and GWNMTF. However, we can see that EKT performs better than WNMTF-TL in all cases. The possible reason is that when the target data is extremely sparse, the collective knowledge transfer way used by EKT may perform better than the adaptive knowledge transfer way used by WNMTF-TL, since the collective behavior can introduce richer interactions when bridging two data sources [31].

4.4 Complexity analysis

The total time complexity of the proposed EKT method is $O(K(MNd_1 + MNd_2 + (M + N)d_1d_2+M^2d_1+N^2d_2)+M^2N+NM^2)$ if the time cost of the similarity graph construction is taken into account, It can be greatly reduced if the input data is sparse. We have also listed in Table 5 the time taken for each method to complete the task of target data sparsity level of 0.1% on the Netflix dataset. This experiment is conducted on a computer 16-GB memory and 1.6-GHz Intel Core i5. From Table 5, we can observe that the non-transfer methods are faster, because they are generally simpler and do not need to consider auxiliary

Table 5 Elapsed time Comparison on the subset of Netflix

Method	Non-transfer			Cross-domain					
	WNMTF	GWNMTF	PMF	TCF(CMTF)	TCF(CSVD)	iTCF	TMF	WNMTF-TL	EKT
Time(s)	4.56	27.47	14.01	116.28	28.87	33.78	43.31	694.27	174.17

data. Among the cross-domain methods, WNMTF-TL is the slowest, because it needs to solve two objective functions, and it also needs to calculate the similarity matrix of entities (users/items).

4.5 Parameter analysis

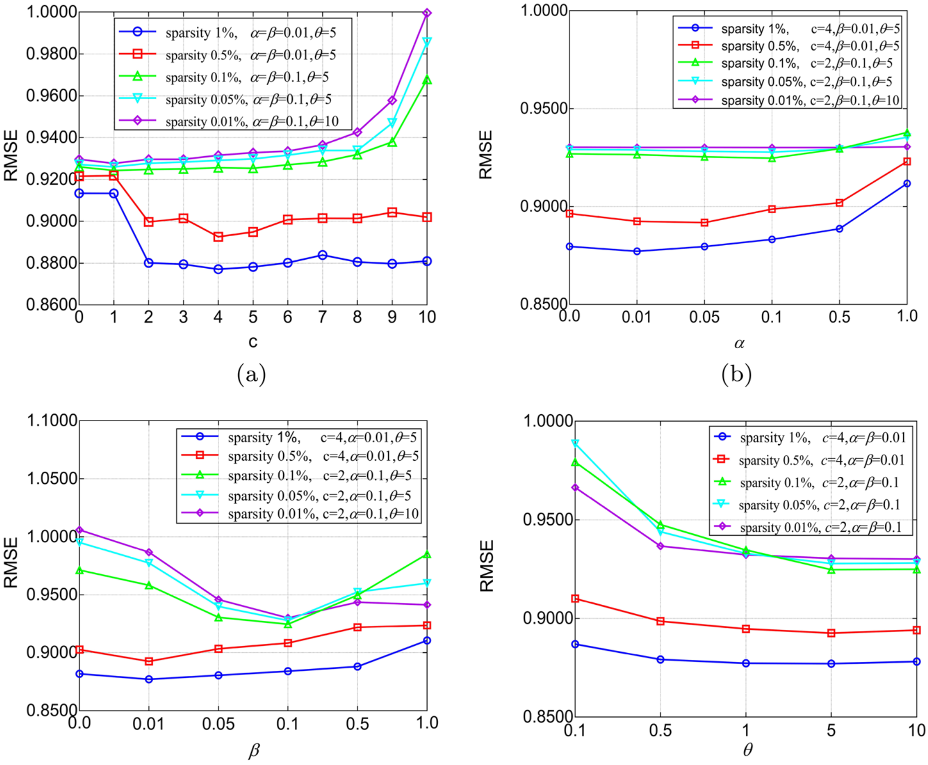
In this section, we test how the parameters affect the performance of EKT. There are 13 parameters in the proposed EKT: λ, d_1, d_2, c, α_u, α_v, β_u, β_v, θ_u, θ_v, α_s, β_s and γ_s. λ is a tradeoff parameter that balances the target data and auxiliary data. d_1 and d_2 are the latent feature number of users and items, respectively. c is the shared rating pattern number. α_u and α_v are the graph regularization parameters of target data. β_u are β_v are the graph regularization parameters of auxiliary data. α_s, β_s and γ_s are the regularization parameters. To simplify the problem, we fixed the parameters $\lambda = 1$, $\alpha_s = \beta_s = \gamma_s = 1$, $d_1 = d_2 = 10$ and let $\alpha_u = \alpha_v = \alpha$, $\beta_u = \beta_v = \beta$, $\theta_u = \theta_v = \theta$, focusing only on how the parameters c, α, β and θ affect the performance of EKT. For simplicity, only the result for the subset of Netflix has been included. Data sets with five sparsity levels are used to test the four parameters. MAE and RMSE are used as evaluation metrics. Since MAE and RMSE are similar, we only show the results of RMSE.

For the parameter c, we search for the grid $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ to find the best parameter value. For the parameters α and β, we search for the grid $\{0.01, 0.05, 0.1, 0.5, 1\}$ to find the best parameter value. For the parameter θ, we search for the grid $\{0.1, 0.5, 1, 5, 10\}$ to find the best parameter value. The best parameter settings for these four parameters at different sparsity levels are listed in Table 6. To analyze the parameter c, we fix α, β and θ to the best parameter values for different sparsity levels, as shown in Table 6. In Fig. 3a, we can see that for target data with different sparsity levels, setting the proper c value can help improve the accuracy of the prediction. For target data with sparsity levels of 1% and 0.5%, the highest accuracy can be achieved by setting $c = 4$, and for target data with sparsity levels of 0.1%, 0.05% and 0.01%, the highest accuracy can be obtained by setting $c = 1$. That is, when the target data is more sparse, the number of rating patterns shared by auxiliary data and target data is less, therefore c should be set smaller to avoid negative transfer. Likewise, to analyze the parameter α, β or θ, we fix the remaining parameters to the best parameter values for different sparsity levels, as shown in Table 6. In Fig. 3b and c, we can observe that when the target data sparsity is higher than 0.1%, better performance improvement can be achieved by setting the proper α ($\alpha = 0.01$) and β ($\beta = 0.01$). However, when the target data sparsity is equal to or lower than 0.1%, α has little effect on the performance improvement, and β has a greater impact on prediction performance. The reason is that when the target data is extremely sparse, the neighborhood structure information in the target data is difficult to be accurately obtained. At this time, by setting with the proper β ($\beta = 0.1$), we can see that the prediction performance can be improved. A reasonable explanation is that the neighborhood structure information of the auxiliary data is

Table 6 The best parameter setting of c, α, β and θ at different sparsity levels

Data set	Parameters	Sparsity				
		0.01%	0.05%	0.1%	0.5%	1%
The subset	c	2	2	2	4	4
of Netflix	α	0.1	0.1	0.1	0.01	0.01
	β	0.1	0.1	0.1	0.01	0.01
	θ	10	5	5	5	5

transferred to the target data to help refine the latent factors of the target data. In Fig. 3b, we can see that the sparser the target data is, a larger θ is required to obtain better prediction performance. The reason is that the sparser the target data needs to transfer more knowledge from the auxiliary data. However, when choosing a higher θ value, it will take more time to run the algorithm. To tradeoff between an acceptable running time for the algorithm and relatively high predictive performance, in our experiments, we set $\theta = 5$ (sparity$\geq$ 0.05%) and $\theta = 10$ (sparity= 0.01%).



(a) (b) (c) (d)

Fig. 3 Result of RMSE with different parameter settings on the subset of Netflix

5 Conclusions and future work

We proposed an Enhanced Knowledge Transfer for Collaborative Filtering with Multi-Source Heterogeneous Feedbacks (EKT), which transfers more useful knowledge from auxiliary data with explicit binary ratings, reducing the sparsity of the target numerical data. By sharing latent user preferences, latent item feature and partial user-item rating pattern between target data and auxiliary data, EKT method can achieve more complete knowledge transfer, while alleviating negative transfer issue by integrating graph co-regularization of user and item graphs into the weighted collective matrix tri-factorization. Experimental results on two benchmark datasets verify the effectiveness of the presented EKT method. And in the case of extremely sparse target data, our EKT method can still achieve relatively good prediction performance.

For future work, our main interest is to extend our EKT method to heterogeneous feedback scenarios with multiple heterogeneous auxiliary sources.

Acknowledgements This work was supported in part by the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (71421001), and in part by the National Natural Science Foundation of China (61772111).

References

1. Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng 17(6):734–749
2. Belkin M, Niyogi P (2002) Laplacian eigenmaps and spectral techniques for embedding and clustering. In: Proc 16th con Adv Neural inform Process Syst (NIPS), Vancouver, British Columbia, Canada, pp 585–591
3. Bell RM, Koren Y (2007) Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In: Proc 7th IEEE int Conf Data min (ICDM), Omaha Nebraska, vol 7, pp 43–52
4. Cai D, He X, Han J, Huang TS (2010) Graph regularized nonnegative matrix factorization for data representation. IEEE Trans Pattern Anal Mach Intell 33(8):1548–1560
5. Cai D, He X, Wang X, Bao H, Han J (2009) Locality preserving nonnegative matrix factorization. In: Proc 21st int Joint conf Artif Intell (IJCAI), Pasadena, California, USA, pp 1010–1015
6. Chen G, Wang F, Zhang C (2009) Collaborative filtering using orthogonal nonnegative matrix tri-factorization. Inform Process Manag 45(3):368–379
7. Chen J, Zhang H, He X, Nie L, Liu W, Chua TS (2017) Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In: Proc 40th int ACM SIGIR conf Res Dev Inform Retrieval, Tokyo, Japan, pp 335–344
8. Covington P, Adams J, Sargin E (2016) Deep neural networks for youtube recommendations. In: Proc 10th ACM conf Rec Syst (recsys), Boston,USA, pp 191–198
9. Deshpande M, Karypis G (2004) Item-based top-n recommendation algorithms. ACM Trans Inform Syst (TOIS) 22(1):143–177
10. Gao S, Luo H, Chen D, Li S, Gallinari P, Guo J (2013) Cross-domain recommendation via cluster-level latent factor model. In: Proc Eur Conf Mach Learn Knowl Disc Databases (ECML/PKDD), Prague, Czech Republic, pp 161–176
11. Gu Q, Zhou J, Ding C (2010) Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs. In: Proc 10th SIAM int Conf on data mining(SDM), Columbus, USA, pp 199–210
12. Hao P, Zhang G, Martinez L, Lu J (2017) Regularizing knowledge transfer in recommendation with tag-inferred correlation. IEEE Trans Cybern 49(1):83–96
13. He X, Liao L, Zhang H, Nie L, Hu X, Chua TS (2017) Neural collaborative filtering. In: Proc 26th int Conf World wide web(WWW), Perth, Western Australia, Australia, pp 173–182
14. Hernando A, Bobadilla J, Ortega F (2016) A non negative matrix factorization for collaborative filtering recommender systems based on a bayesian probabilistic model. Knowl-Based Syst 97:188–202

15. Hu G, Zhang Y, Yang Q (2018) Conet: Collaborative cross networks for cross-domain recommendation. In: Proc 27th ACM int Conf Inform Knowl Manag (CIKM), Turin, Italy, pp 667–676
16. Lee DD, Seung HS (2001) Algorithms for non-negative matrix factorization. In: Proc 15th adv Neural inform Process Syst (NIPS), Vancouver, British Columbia, Canada, pp 556–562
17. Li B (2011) Cross-domain collaborative filtering: A brief survey. In: Proc. 23rd IEEE int Conf Tools artif Intell (ICTAI), Boca Raton, Florida, USA, pp 1085–1086
18. Li B, Yang Q, Xue X (2009) Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. In: Proc 21st int Joint conf Artif Intell (IJCAI), Pasadena, California, USA, pp 2052–2057
19. Li B, Yang Q, Xue X (2009) Transfer learning for collaborative filtering via a rating-matrix generative model. In: Proc 26th annu Int Conf Mach Learn (ICML), Montreal, QC, Canada, pp 617–624
20. Li B, Zhu X, Li R, Zhang C (2014) Rating knowledge sharing in cross-domain collaborative filtering. IEEE Trans Cybern 45(5):1068–1082
21. Long M, Wang J, Ding G, Shen D, Yang Q (2013) Transfer learning with graph co-regularization. IEEE Trans Knowl Data Eng 26(7):1805–1818
22. Lu Z, Zhong E, Zhao L, Xiang EW, Pan W, Yang Q (2013) Selective transfer learning for cross domain recommendation. In: Proc 13th SIAM int Conf Data mining(SDM), Austin, Texas, USA, pp 641–649
23. Ma H, Yang H, Lyu MR, King I (2008) Sorec: Social recommendation using probabilistic matrix factorization. In: Proc 17th ACM conf Inform Knowl Manag (CIKM), Napa, California, USA, pp 931–940
24. Mnih A, Salakhutdinov RR (2008) Probabilistic matrix factorization. In: Proc Adv Neural inform Process Syst (NIPS), Vancouver, Canada, pp 1257–1264
25. Pan SJ, Yang Q (2009) A survey on transfer learning. IEEE Trans Knowl Data Eng 22(10):1345–1359
26. Pan W (2016) A survey of transfer learning for collaborative recommendation with auxiliary data. Neurocomputing 177:447–453
27. Pan W, Liu NN, Xiang EW, Yang Q (2011) Transfer learning to predict missing ratings via heterogeneous user feedbacks. In: Proc 22nd int Joint conf Artif Intell (IJCAI), Barcelona, Catalonia, Spain, pp 2318–2323
28. Pan W, Ming Z (2014) Interaction-rich transfer learning for collaborative filtering with heterogeneous user feedback. IEEE Intell Syst 29(6):48–54
29. Pan W, Xia S, Liu Z, Peng X, Ming Z (2016) Mixed factorization for collaborative recommendation with heterogeneous explicit feedbacks. Inform Sci 332:84–93
30. Pan W, Xiang EW, Liu NN, Yang Q (2010) Transfer learning in collaborative filtering for sparsity reduction. In: 230–235
31. Pan W, Yang Q (2013) Transfer learning in heterogeneous collaborative filtering domains. Artif Intell 197:39–55
32. Sarwar BM, Karypis G, Konstan JA, Riedl J et al (2001) Item-based collaborative filtering recommendation algorithms. In: Proc 10th int Conf World wide web (WWW), Hong Kong, China, vol 1, pp 285–295
33. Shi J, Long M, Liu Q, Ding G, Wang J (2013) Twin bridge transfer learning for sparse collaborative filtering. In: Pacific-asia conf Knowl Discov Data mining, Gold Coast, Australia, pp 496–507
34. Sindhwani V, Bucak S, Hu J, Mojsilovic A (2009) A family of non-negative matrix factorizations for one-class collaborative filtering problems. In: Proc 3rd ACM conf Rec Syst, (recsys). New york, NY, USA
35. Singh AP, Gordon GJ (2008) Relational learning via collective matrix factorization. In: Proc. 14th ACM SIGKDD int Conf Knowl Discov Data mining(KDD), Las Vegas, NV, USA, pp 650–658
36. Smith B, Linden G (2017) Two decades of recommender systems at amazon. com. IEEE Int Comput 21(3):12–18
37. Srebro N, Rennie J, Jaakkola TS (2005) Maximum-margin matrix factorization. In: Proc 19th conf Adv Neural inform Process Syst (NIPS), Vancouver, British Columbia, Canada, pp 1329–1336
38. Su X, Khoshgoftaar TM (2009) A survey of collaborative filtering techniques. Adv Artif Intell 2009(4)
39. Wu L, Sun P, Hong R, Ge Y, Wang M (2018) Collaborative neural social recommendation. IEEE Trans Syst Man Cybern Syst 1–13
40. Yang B, Lei Y, Liu J, Li W (2016) Social collaborative filtering by trust. IEEE Trans Pattern Anal Mach Intell 39(8):1633–1647
41. Zhang H, Ni W, Li X, Yang Y (2018) Modeling the heterogeneous duration of user interest in time-dependent recommendation: A hidden semi-markov approach. IEEE Trans Syst Man Cybern Syst 48(2):177–194
42. Zhang M, Guo X, Chen G (2016) Prediction uncertainty in collaborative filtering: Enhancing personalized online product ranking. Decis Support Syst 83:10–21

43. Zhang Q, Lu J, Wu D, Zhang G (2019) A cross-domain recommender system with kernel-induced knowledge transfer for overlapping entities. IEEE Trans Neural Netw 30(7):1998–2012
44. Zhang Q, Wu D, Lu J, Liu F, Zhang G (2017) A cross-domain recommender system with consistent information transfer. Decis Support Syst 104:49–63
45. Zhang S, Wang W, Ford J, Makedon F (2006) Learning from incomplete ratings using non-negative matrix factorization. In: Proc 6th SIAM int Conf Data min (SDM), Bethesda, MD, USA, pp 549–553
46. Zhao L, Pan SJ, Yang Q (2017) A unified framework of active transfer learning for cross-system recommendation. Artif Intell 245:38–55

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Hongwei Zhang[1,2] **· Xiangwei Kong**[3] **· Yujia Zhang**[4]

Hongwei Zhang
hwzhang82@mail.dlut.edu.cn

Yujia Zhang
yjzhang7@seas.upenn.edu

[1] School of Information and Communication Engineering, Dalian University of Technology, Dalian, 116024, China

[2] School of Mathematics, Tonghua Normal University, Tonghua, 134002, China

[3] Department of Data Science and Engineering Management, Zhejiang University, Hangzhou, 310058, China

[4] School of Engineering and Applied Science, University of Pennsylvania, Philadelphia, PA, 19104-6391, USA