

Twin Bridge Transfer Learning for Sparse Collaborative Filtering

Jiangfeng Shi^{1,2}, Mingsheng Long^{1,2}, Qiang Liu¹,
Guiguang Ding¹, and Jianmin Wang¹

¹ MOE Key Laboratory for Information System Security;
TNLIST; School of Software

² Department of Computer Science & Technology,
Tsinghua University, Beijing, China

{shijiangfengsjf, longmingsheng}@gmail.com,
{liuqiang, dinggg, jimwang}@tsinghua.edu.cn

Abstract. Collaborative filtering (CF) is widely applied in recommender systems. However, the *sparsity* issue is still a crucial bottleneck for most existing CF methods. Although target data are extremely sparse for a newly-built CF system, some dense auxiliary data may already exist in other matured related domains. In this paper, we propose a novel approach, *Twin Bridge Transfer Learning* (TBT), to address the sparse collaborative filtering problem. TBT reduces the sparsity in target data by transferring knowledge from dense auxiliary data through two paths: 1) the *latent factors* of users and items learned from two dense auxiliary domains, and 2) the *similarity graphs* of users and items constructed from the learned latent factors. These two paths act as a *twin bridge* to allow more knowledge transferred across domains to reduce the sparsity of target data. Experiments on two benchmark datasets demonstrate that our TBT approach significantly outperforms state-of-the-art CF methods.

Keywords: Collaborative Filtering, Sparsity, Transfer Learning, Latent Factor, Similarity Graph, Twin Bridge.

1 Introduction

Recommender systems are widely deployed on the Web with the aim to improve the user experience [1]. Well-known e-commerce providers, such as Hulu (video) and Netflix (movies), are highly relying on their developed recommender systems. These systems attempt to recommend items that are most likely to attract users by predicting the preference (ratings) of the users to the items. In the last decade, *collaborative filtering* (CF) methods [2–4], which utilize users' past ratings to predict users' future tastes, have been most popular due to their generally superior performance [5, 6]. Unfortunately, most of previous CF methods have not fully addressed the sparsity issue, i.e., the target data are extremely sparse.

Recently, there has been an increasing research interest in developed *transfer learning* methods to alleviate the data sparsity problem [7–13]. The key idea behind is to transfer the *common knowledge* from some dense auxiliary data

to the sparse target data. The common knowledge can be encoded into some common latent factors, and extracted by the latent factorization models [14]. Good choices for the common latent factors can be the cluster-level codebook [8, 9], or the latent tastes of users/latent features of items [11–13]. The transfer learning methods have achieved promising performance over previous methods.

However, when the target data are *extremely* sparse, existing transfer learning methods may confront the following issues. 1) Since the target data are extremely sparse, the latent factors extracted from the auxiliary data may transfer *negative* information to the target data, i.e., the latent factors are likely to be inconsistent with the target data, which may result in overfitting [15]. We refer to this issue as *negative transfer*. 2) As the target data are extremely sparse, it is expected that more common knowledge should be transferred from the auxiliary data to reduce the sparsity of the target data [16]. We refer to this issue as *insufficient transfer*. Solving these two issues both can reduce the sparsity more effectively.

In this paper, we propose a novel approach, *Twin Bridge Transfer Learning* (TBT), for sparse collaborative filtering. TBT aims to explore a *twin bridge* corresponding to the *latent factors* which encode the latent tastes of users/latent features of items, and the *similarity graphs* which encode the collaboration information between users/items. TBT consists of two sequential steps. 1) It extracts the latent factors from dense auxiliary data and constructs the similarity graphs from these extracted latent factors. 2) It transfers both the latent factors and the similarity graphs to the target data. In this way, we can enhance sufficient transfer by transferring more common knowledge between domains, and alleviate negative transfer by filtering out the negative information introduced in the latent factors. Extensive experiments on two real-world data sets show promising results obtained by TBT, especially when the target data are extremely sparse.

2 Related Work

Prior works using transfer learning for collaborative filtering assume that there exist a set of common latent factors between the auxiliary and target data. Rating Matrix Generative Model (RMGM) [9] and Codebook Transfer (CBT) [8] transfer the cluster-level codebook to the target data. However, since codebook dimension cannot be too large, the codebook cannot transfer enough knowledge when the target data are extremely sparse. To avoid this limitation, Coordinate System Transfer (CST) [12], Transfer by Collective Factorization (TCF) [11], and Transfer by Integrative Factorization (TIF) [13] extract both latent tastes of users and latent features of items, and transfer them to the target data. However, these methods do not consider the negative transfer issue, which is more serious when the target data are extremely sparse. Also, they only utilize a single bridge for knowledge transfer. Different from all these methods, our approach explores a twin bridge for transferring more useful knowledge to the sparse target data.

Neighborhood structure has been widely used in memory-based CF [17] or graph regularized methods [4]. Neighborhood-Based CF Algorithm [17] predicts missing ratings of target users by their nearest neighbors. Graph Regularized Weighted Nonnegative Matrix Factorization (GWNMTF) [4] encodes the neighborhood

Table 1. Notations and their descriptions in this paper

Notation	Description	Notation	Description
τ	dataset index, $\tau \in \{1, 2\}$	\mathbf{R}	$m \times n$ rating matrix
m	#target users	\mathbf{Z}	$m \times n$ indicator matrix
n	#target items	\mathbf{U}	$m \times k$ latent tastes of users
p	#nearest neighbors	\mathbf{V}	$n \times k$ latent features of items
k	#latent factors	\mathbf{B}	$k \times k$ cluster-level codebook
λ_U, λ_V	regularization param of latent factors	γ_U, γ_V	regularization param of similarity graphs

information into latent factor models, which can preserve the similarity between user tastes or item features. However, these methods depend largely on dense ratings to compute accurate neighborhood structure, which is impossible when the data are extremely sparse. Different from these methods, our approach extracts the latent factors of users/items from some dense auxiliary data, and then constructs the neighborhood structure from these latent factors. In this way, the constructed neighborhood structure is more accurate for sparse CF.

3 Problem Definition

We focus on *sparse collaborative filtering* where the target data are *extremely* sparse. Suppose in the target data, we have m users and n items, and a set of integer ratings ranging from 1 to 5. $\mathbf{R} \in \mathbb{R}^{m \times n}$ is the rating matrix, where R_{ij} is the rating that user i gives to item j . $\mathbf{Z} \in \{0, 1\}$ is the indicator matrix, $Z_{ij} = 1$ if user i has rated item j , and $Z_{ij} = 0$ otherwise. In order to reduce the sparsity in the target data, we assume that there exist some dense auxiliary data, from which we can transfer some common knowledge to the target data. Specifically, we will consider two sets of auxiliary data, whose rating matrices are \mathbf{R}_1 and \mathbf{R}_2 , with indicator matrices \mathbf{Z}_1 and \mathbf{Z}_2 , respectively. \mathbf{R}_1 shares the common set of users with \mathbf{R} , while \mathbf{R}_2 shares the common set of items with \mathbf{R} . For clarity, the notations and their descriptions are summarized in Table 1.

Definition 1 (Learning Goal). *The goal of TBT is to construct 1) latent factors $\mathbf{U}_0, \mathbf{V}_0$ from $\mathbf{R}_1, \mathbf{R}_2$ and 2) similarity graphs $\mathbf{L}_U, \mathbf{L}_V$ from $\mathbf{U}_0, \mathbf{V}_0$ (Figure 1), and use them as the twin bridge to predict missing values in \mathbf{R} (Figure 2).*

4 Twin Bridge Transfer Learning for Recommendation

In this section, we present our proposed TBT approach, which is comprised of two sequential steps: 1) twin bridge construction, and 2) twin bridge transfer.

4.1 Twin Bridge Construction

In the first step, we aim to construct the latent factors and similarity graphs of users and items, which will be used as the twin bridge for knowledge transfer.

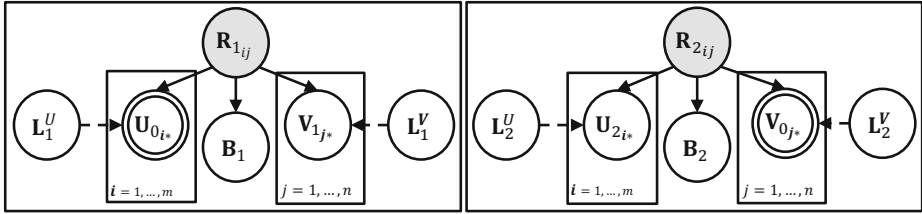


Fig. 1. Step 1 of TBT: twin bridge construction. The goal is to extract the latent factors and construct the similarity graphs from two sets of dense auxiliary data.

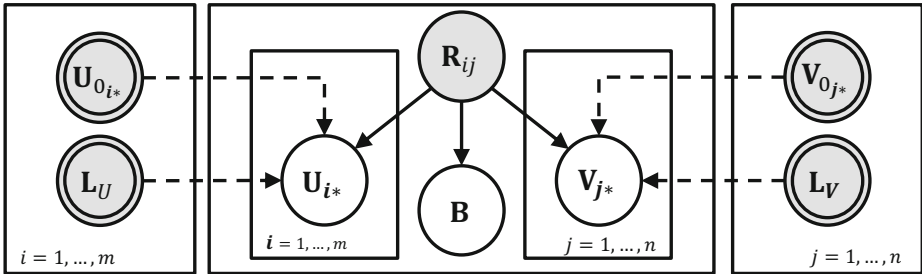


Fig. 2. Step 2 of TBT: twin bridge transfer. '→' represents matrix tri-factorization, '− →' represents twin bridge transfer, ⊙ represents the transferred knowledge structure

Latent Factor Extraction. We adopt the *graph regularized weighted nonnegative matrix tri-factorization* (GWNMTF) method proposed by Gu et al. [4] to extract the latent factors from the two sets of dense auxiliary data \mathbf{R}_1 and \mathbf{R}_2

$$\mathcal{L} = \min_{\mathbf{U}_\tau, \mathbf{B}_\tau, \mathbf{V}_\tau \geq 0} \left\| \mathbf{Z}_\tau \odot (\mathbf{R}_\tau - \mathbf{U}_\tau \mathbf{B}_\tau \mathbf{V}_\tau^T) \right\|_F^2 + \gamma_U \text{tr}(\mathbf{U}_\tau^T \mathbf{L}_\tau^U \mathbf{U}_\tau) + \gamma_V \text{tr}(\mathbf{V}_\tau^T \mathbf{L}_\tau^V \mathbf{V}_\tau) \tag{1}$$

where γ_U and γ_V are graph regularization parameters, $\mathbf{L}_\tau^U, \mathbf{L}_\tau^V$ are the graph Laplacian matrices, $\|\cdot\|$ is the Frobenius norm of matrix. $\mathbf{U}_\tau = [u_{*1}^\tau, \dots, u_{*k}^\tau] \in \mathbb{R}^{m \times k}$ are the k latent tastes of users, with each u_{i*}^τ representing the latent taste of user i . $\mathbf{V}_\tau = [v_{*1}^\tau, \dots, v_{*k}^\tau] \in \mathbb{R}^{n \times k}$ are the k latent features of items, with each v_{i*}^τ representing the latent feature of item i . $\mathbf{B}_\tau \in \mathbb{R}^{k \times k}$ is the cluster-level codebook representing the association between \mathbf{U}_τ and $\mathbf{V}_\tau, \tau \in \{1, 2\}$.

Since auxiliary data \mathbf{R}_1 share the common set of users with the target data \mathbf{R} , we can share the latent tastes of users between them, that is, $\mathbf{U}_0 = \mathbf{U}_1$, where \mathbf{U}_0 denotes the common latent tastes of users. Similarly, since auxiliary data \mathbf{R}_2 share the common set of items with the target data \mathbf{R} , we can share the latent features of items between them, that is, $\mathbf{V}_0 = \mathbf{V}_2$, where \mathbf{V}_0 denotes the common latent features of items. \mathbf{U}_0 and \mathbf{V}_0 are the *common latent factors*, which will be used as the first type of bridge for knowledge transfer.

It is worth noting that, the latent factors extracted by GWNMTF can respect the intrinsic neighborhood structure better than those extracted by Sparse Singular Value Decomposition (SVD) [12], thus can fit sparse target data better.

Similarity Graph Construction. Collaboration information between users or items contained in the rating matrix has been widely explored for CF. However, in extremely sparse target data, two users may rate the same item with low probability though they have the same taste. To avoid this data sparsity issue, we construct the similarity graphs from dense auxiliary data since they share common users or items with target data. Because \mathbf{U}_0 and \mathbf{V}_0 are denser than \mathbf{R}_1 and \mathbf{R}_2 and can better respect the collaboration information underlying the target data, we propose to construct similarity graphs of users and items from \mathbf{U}_0 and \mathbf{V}_0 , respectively. We define the distance between users/items as follows:

$$\begin{aligned} (\mathbf{W}_U)_{ij} &= \begin{cases} 1, & \text{if } \mathbf{u}_{i*}^0 \in \mathcal{N}_p(\mathbf{u}_{j*}^0) \text{ or } \mathbf{u}_{j*}^0 \in \mathcal{N}_p(\mathbf{u}_{i*}^0) \\ 0, & \text{otherwise} \end{cases} \\ (\mathbf{W}_V)_{ij} &= \begin{cases} 1, & \text{if } \mathbf{v}_{i*}^0 \in \mathcal{N}_p(\mathbf{v}_{j*}^0) \text{ or } \mathbf{v}_{j*}^0 \in \mathcal{N}_p(\mathbf{v}_{i*}^0) \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (2)$$

where \mathbf{u}_{i*}^0 is the i th row of \mathbf{U}_0 to denote the latent taste of user i , \mathbf{v}_{i*}^0 is the i th row of \mathbf{V}_0 to denote the latent feature of item i , $\mathcal{N}_p(\mathbf{u}_{i*}^0)$ and $\mathcal{N}_p(\mathbf{v}_{i*}^0)$ are the sets of p -nearest neighbors of \mathbf{u}_{i*}^0 and \mathbf{v}_{i*}^0 , respectively. The positive semi-definite symmetric matrices \mathbf{W}_U and \mathbf{W}_V are the weight matrices for the *similarity graphs*, which will be used as the second type of bridge for knowledge transfer.

4.2 Twin Bridge Transfer

In the second step, we introduce the objectives for latent factor transfer and similarity graph transfer respectively, and then integrate them into a unified optimization problem for twin bridge transfer learning.

Latent Factor Transfer. Since the latent factors of users/items may not be shared as a whole between domains, we propose to transfer latent factors $\mathbf{U}_0, \mathbf{V}_0$ to the target data via two regularization terms $\|\mathbf{U} - \mathbf{U}_0\|_F^2$ and $\|\mathbf{V} - \mathbf{V}_0\|_F^2$, instead of enforcing $\mathbf{U} \equiv \mathbf{U}_0, \mathbf{V} \equiv \mathbf{V}_0$. This leads to the *latent factor transfer*

$$\min_{\mathbf{U}, \mathbf{V}, \mathbf{B} \geq 0} \mathcal{O}_{UV} = \|\mathbf{Z} \odot (\mathbf{R} - \mathbf{UBV}^T)\|_F^2 + \lambda_U \|\mathbf{U} - \mathbf{U}_0\|_F^2 + \lambda_V \|\mathbf{V} - \mathbf{V}_0\|_F^2 \quad (3)$$

where λ_U, λ_V are regularization parameters indicating our confidence on the latent factors. Since the target data are extremely sparse, the latent factors $\mathbf{U}_0, \mathbf{V}_0$ may transfer some negative information to the target data. Therefore, we introduce similarity graph transfer as a complementary method in the sequel.

Similarity Graph Transfer. Based on the *manifold assumption* [18], similar users/items should have similar latent tastes/features. Therefore, preserving the neighborhood structure underlying the users/items in the target data are reduced to the following graph regularizers encoding the constructed similarity graphs

$$\begin{aligned}\mathcal{G}_U &= \frac{1}{2} \sum_{ij} \|\mathbf{u}_{i*} - \mathbf{u}_{j*}\|^2 (\mathbf{W}_U)_{ij} = \text{tr}(\mathbf{U}^T (\mathbf{D}_U - \mathbf{W}_U) \mathbf{U}) = \text{tr}(\mathbf{U}^T \mathbf{L}_U \mathbf{U}) \\ \mathcal{G}_V &= \frac{1}{2} \sum_{ij} \|\mathbf{v}_{i*} - \mathbf{v}_{j*}\|^2 (\mathbf{W}_V)_{ij} = \text{tr}(\mathbf{V}^T (\mathbf{D}_V - \mathbf{W}_V) \mathbf{V}) = \text{tr}(\mathbf{V}^T \mathbf{L}_V \mathbf{V})\end{aligned}\quad (4)$$

where $\mathbf{D}_U = \text{diag}(\sum_j (\mathbf{W}_U)_{ij})$, $\mathbf{D}_V = \text{diag}(\sum_j (\mathbf{W}_V)_{ij})$. $\mathbf{L}_U = \mathbf{D}_U - \mathbf{W}_U$ and $\mathbf{L}_V = \mathbf{D}_V - \mathbf{W}_V$ are the graph Laplacian matrices for the similarity graphs.

Incorporating the graph regularizers into the objective function of nonnegative matrix tri-factorization, we obtain the following *similarity graph transfer*

$$\min_{\mathbf{U}, \mathbf{V}, \mathbf{B} \geq 0} \mathcal{O}_{\mathcal{G}_U \mathcal{G}_V} = \|\mathbf{Z} \odot (\mathbf{R} - \mathbf{UBV}^T)\|_F^2 + \gamma_U \mathcal{G}_U + \gamma_V \mathcal{G}_V \quad (5)$$

where γ_U, γ_V are regularization parameters indicating our confidence on the similarity graphs. With similarity graph transfer, we can essentially filter out the negative information in the latent factors by using the nearest neighbor rule. However, since the target data are extremely sparse, only transferring knowledge from the similarity graphs may be insufficient. Thus we seek an integrated model.

Twin Bridge Transfer. Considering the aforementioned limitations, we integrate the latent factor transfer and similarity graph transfer into a unified *twin bridge transfer* (TBT) method. In TBT, firstly, we extract the latent factors of users/items and construct the similarity graphs from them. Secondly, we transfer latent factors as the first bridge to solve the insufficient transfer issue. As mentioned before, these latent factors may contain negative information for the target data and result in the negative transfer issue. So thirdly, we transfer the similarity graphs as the second bridge to alleviate negative transfer and to transfer more knowledge to the target data. It is worth noting that, each of the twin bridge can enhance the learning of the other bridge during the optimization procedure due to their complementary property. Therefore, we incorporate Equations (3) and (5) into a unified *Twin Bridge Transfer* optimization problem

$$\min_{\mathbf{U}, \mathbf{V}, \mathbf{B} \geq 0} \mathcal{O} = \|\mathbf{Z} \odot (\mathbf{R} - \mathbf{UBV}^T)\|_F^2 + \lambda_U \|\mathbf{U} - \mathbf{U}_0\|_F^2 + \lambda_V \|\mathbf{V} - \mathbf{V}_0\|_F^2 + \gamma_U \mathcal{G}_U + \gamma_V \mathcal{G}_V \quad (6)$$

where $\lambda_U, \lambda_V, \gamma_U, \gamma_V$ are regularization parameters. If $\lambda_U, \lambda_V > \gamma_U, \gamma_V$, the transferred latent factors dominate the recommendation results; otherwise, the transferred similarity graphs dominate the recommendation results. With TBT, we can transfer more knowledge from dense auxiliary data to sparse target data to reduce the data sparsity, and substantially avoid the negative transfer issue.

4.3 Learning Algorithm

We solve the optimization problem in Equation (6) using the gradient descent method. The derivative of \mathcal{O} with respect to \mathbf{U} (the other variables are fixed) is

$$\frac{\partial \mathcal{O}}{\partial \mathbf{U}} = -2\mathbf{Z} \odot \mathbf{R}\mathbf{V}\mathbf{B}^T + 2\mathbf{Z} \odot (\mathbf{UBV}^T) \mathbf{V}\mathbf{B}^T + 2\lambda_U (\mathbf{U} - \mathbf{U}_0) + 2\gamma_U \mathbf{L}_U \mathbf{U}$$

Algorithm 1. TBT: Twin Bridge Transfer Learning for Collaborative Filtering**Input:** Data sets $\mathbf{R}, \mathbf{R}_\tau, \mathbf{Z}, \mathbf{Z}_\tau, \tau \in \{1, 2\}$, parameters $k, p, \lambda_U, \lambda_V, \gamma_U, \gamma_V$.**Output:** Latent factors \mathbf{U}, \mathbf{V} , and \mathbf{B} .

- 1: Extract latent factor $\mathbf{U}_0, \mathbf{V}_0$ by Equations (1).
- 2: Construct similarity graphs $\mathcal{G}_U, \mathcal{G}_V$ from $\mathbf{U}_0, \mathbf{V}_0$ by Equations (4).
- 3: **for** $it \leftarrow 1$ to $maxIt$ **do**
- 4: update \mathbf{U}, \mathbf{V} , and \mathbf{B} by Equations (7)~(9), respectively.
- 5: **end for**

Using Karush-Kuhn-Tucker (KKT) condition [19] for nonnegativity of \mathbf{U} , we get

$$[-\mathbf{Z} \odot \mathbf{R}\mathbf{V}\mathbf{B}^T + \mathbf{Z} \odot (\mathbf{U}\mathbf{B}\mathbf{V}^T) \mathbf{V}\mathbf{B}^T + \lambda_U(\mathbf{U} - \mathbf{U}_0) + \gamma_U \mathbf{L}_U \mathbf{U}] \odot \mathbf{U} = 0$$

Since \mathbf{L}_U may take mixed signs, we replace it with $\mathbf{L}_U = \mathbf{L}_U^+ - \mathbf{L}_U^-$, where $\mathbf{L}_U^+ = \frac{1}{2}(|\mathbf{L}_U| + \mathbf{L}_U)$, $\mathbf{L}_U^- = \frac{1}{2}(|\mathbf{L}_U| - \mathbf{L}_U)$ are positive-valued matrices. We obtain

$$\mathbf{U} \leftarrow \mathbf{U} \odot \sqrt{\frac{\mathbf{Z} \odot \mathbf{R}\mathbf{V}\mathbf{B}^T + \lambda_U \mathbf{U}_0 + \gamma_U \mathbf{L}_U^- \mathbf{U}}{\mathbf{Z} \odot (\mathbf{U}\mathbf{B}\mathbf{V}^T) \mathbf{V}\mathbf{B}^T + \lambda_U \mathbf{U} + \gamma_U \mathbf{L}_U^+ \mathbf{U}}} \quad (7)$$

Likewise, we derive \mathbf{V} and \mathbf{B} in similar way and get the following update rules

$$\mathbf{V} \leftarrow \mathbf{V} \odot \sqrt{\frac{(\mathbf{Z} \odot \mathbf{R})^T \mathbf{U}\mathbf{B} + \lambda_V \mathbf{V}_0 + \gamma_V \mathbf{L}_V^- \mathbf{V}}{(\mathbf{Z} \odot (\mathbf{U}\mathbf{B}\mathbf{V}^T))^T \mathbf{U}\mathbf{B} + \lambda_V \mathbf{V} + \gamma_V \mathbf{L}_V^+ \mathbf{V}}} \quad (8)$$

$$\mathbf{B} \leftarrow \mathbf{B} \odot \sqrt{\frac{\mathbf{U}^T (\mathbf{Z} \odot \mathbf{R}) \mathbf{V}}{\mathbf{U}^T (\mathbf{Z} \odot (\mathbf{U}\mathbf{B}\mathbf{V}^T)) \mathbf{V}}} \quad (9)$$

According to [4], updating \mathbf{U}, \mathbf{V} and \mathbf{B} sequentially by Equations (7)~(9) will monotonically decrease the objective function in Equation (6) until convergence.

The learning algorithm for TBT is summarized in Algorithm 1. The time complexity is $O(maxIt \cdot kmn + mn^2 + m^2n)$, which is the sum of TBT cost plus the time cost for the latent factor extraction and similarity graphs construction.

5 Experiments

In this section, we conduct experiments to compare TBT with state-of-the-art collaborative filtering methods on benchmark data sets with high sparsity level.

5.1 Data Sets

MovieLens10M¹ MovieLens10M consists of 10,000,054 ratings and 95,580 tags given by 71,567 users to 10,681 movies. The preference of the user for a movie is rated from 1 to 5 and 0 indicates that the movie is not rated by any user.

¹ <http://www.grouplens.org/node/73/>

Table 2. Description of the data sets for evaluation

	Data Sets	MovieLens10M		Epinions	
		Size	Sparsity	Size	Sparsity
R	Target (training)	5000×5000	0.01~1%	1662×2078	0.01~1%
	Target (test)		8.22~8.13%		1.86~0.87%
R₁ (user side)	Auxiliary		2.31%		0.65%
R₂ (item side)	Auxiliary		9.14%		1.28%

Epinions² In *epinions.com*, users can assign products with integer ratings ranging from 1 to 5. The dataset used in our experiments is collected by crawling the *epinions.com* website during March, 2012. It consists of 3,324 users who have rated a total of 4,156 items, and the total number of ratings is 138,133.

The data sets used in the experiments are constructed using the same strategy as [12]. For the MovieLens10M data set, we first randomly sample a $10^4 \times 10^4$ dense rating matrix \mathbf{X} from the MovieLens data. Then we take the submatrices $\mathbf{R} = \mathbf{X}_{1 \sim 5000, 1 \sim 5000}$ as the target rating matrix, $\mathbf{R}_1 = \mathbf{X}_{1 \sim 5000, 5001 \sim 10000}$ as the user-side auxiliary data and $\mathbf{R}_2 = \mathbf{X}_{5001 \sim 10000, 1 \sim 5000}$ as the item-side auxiliary data. In this way, \mathbf{R} and \mathbf{R}_1 share common users but not common items, while \mathbf{R} and \mathbf{R}_2 share common items but not common users. We use the same construction strategy for the Epinions data. In all experiments, the target ratings \mathbf{R} are randomly split into a training set and a test set. To investigate the performance of each method on sparse data, we sampled training set randomly with a variety of sparsity levels from 0.01% to 1%, as summarized in Table 2.

5.2 Evaluation Metrics

We adopt two evaluation metrics, *Mean Absolute Error* (MAE) and *Root Mean Square Error* (RMSE), which are widely used for evaluating collaborative filtering algorithms [4, 12]. **MAE** and **RMSE** are defined as

$$MAE = \frac{\sum_{i,j} |R_{ij} - \tilde{R}_{ij}|}{|T_E|} \quad RMSE = \sqrt{\frac{\sum_{i,j} (R_{ij} - \tilde{R}_{ij})^2}{|T_E|}}$$

Where R_{ij} is the rating that user i gives to item j in test set, while \tilde{R}_{ij} is the predicted value of R_{ij} by CF algorithms, $|T_E|$ is the number of ratings in test set. We run each method 10 repeated trials with randomly generated training set and test set from \mathbf{R} , and report the average MAE and RMSE of each method.

5.3 Baseline Methods & Parameter Settings

We compare TBT with *Probabilistic Matrix Factorization* (PMF) [20], *Weight Nonnegative Matrix Factorization* (WNMF) [21], *Graph Regularized Weighted Nonnegative Matrix Tri-Factorization* (GWNMTF) [4], *Coordinate System Transfer* (CST) [12], our proposed **TBT_{UV}** (defined in Equation (3)) and **TBT_G**

² <http://www.epinions.com/>

Table 3. MAE & RMSE comparison on MovieLens10M

	Sparsity	Methods Without Transfer			Method With Transfer			
		PMF	WNMF	GWNMTF	CST	TBT _G	TBT _{UV}	TBT
MAE	0.01%	1.171	2.7966	0.8321	0.9814	0.7535	0.7272	0.6978±0.0006
	0.05%	1.0256	1.3835	0.7663	0.8409	0.6945	0.7492	0.6903±0.0007
	0.10%	0.90232	0.9506	0.7407	0.7629	0.6853	0.7275	0.6840±0.0005
	0.50%	0.73368	0.697	0.6862	0.6799	0.6707	0.6785	0.6690±0.0001
	1%	0.70398	0.6743	0.6728	0.6677	0.6649	0.6677	0.6640±0.0000
	Average	0.9073	1.3004	0.7396	0.7866	0.6938	0.7100	0.6810±0.0004
RMSE	0.01%	1.3717	3.1434	1.0299	1.3926	0.9445	0.9397	0.8995±0.0011
	0.05%	1.2333	1.8779	0.97	1.1225	0.8896	0.9594	0.8893±0.0012
	0.10%	1.1097	1.3199	0.944	0.997	0.8812	0.935	0.8817±0.0008
	0.50%	0.93171	0.9076	0.8802	0.8755	0.8628	0.8734	0.8608±0.0001
	1%	0.90112	0.8718	0.8645	0.8605	0.8561	0.8601	0.8550±0.0000
	Average	1.1095	1.6241	0.9377	1.0496	0.8868	0.9135	0.8772±0.0007

Table 4. MAE & RMSE comparison on Epinions

	Sparsity	Methods Without Transfer			Method With Transfer			
		PMF	WNMF	GWNMTF	CST	TBT _G	TBT _{UV}	TBT
MAE	0.01%	1.6758	3.7617	0.9875	0.9781	0.9066	0.8806	0.8737±0.0046
	0.05%	1.6341	3.0304	0.9792	0.9629	0.8838	0.9122	0.8740±0.0043
	0.10%	1.5555	2.3353	0.9682	0.9506	0.8773	0.9258	0.8682±0.0018
	0.50%	1.1083	1.0465	0.8902	0.8837	0.8543	0.8846	0.8441±0.0016
	1%	0.9824	0.8769	0.8539	0.8467	0.8379	0.8478	0.8294±0.0008
	Average	1.3912	2.2102	0.9358	0.9244	0.8720	0.8902	0.8579±0.0026
RMSE	0.01%	1.8474	3.999	1.2665	1.3063	1.1427	1.1366	1.1164±0.0022
	0.05%	1.8136	3.4811	1.2508	1.2695	1.118	1.1755	1.1116±0.0022
	0.10%	1.7464	2.9094	1.233	1.2403	1.1173	1.1919	1.1105±0.0021
	0.50%	1.3641	1.4222	1.1289	1.1422	1.0885	1.1369	1.0790±0.0007
	1%	1.218	1.1457	1.0932	1.0981	1.0766	1.095	1.0688±0.0005
	Average	1.5979	2.5915	1.1945	1.2113	1.1086	1.1472	1.0973±0.0015

(defined in Equation (5)). Different numbers of latent factors $\{5, 10, 15, 20\}$ and different values of regularization parameters $\{0.01, 0.1, 1, 10, 100\}$ of each method are tried, with the best ones selected for comparison. Since all the algorithms are iterative ones, we run each of them 1000 iterations and report the best results.

5.4 Experimental Results

The experimental results on MovieLens10M and Epinions are shown in Table 3 and Table 4 respectively. From the tables, we can make several observations.

- TBT performs better than all baseline methods under all sparsity levels. It confirms that simultaneously transferring latent factors and similarity graphs as twin bridge can successfully reduce the sparsity in the target data.
- It is interesting to see that, for our method, the sparser the target data are, the more improvement can be achieved. This can be explained by two reasons.

First, when the rating data are sparse, matrix factorization methods that rely on neighborhood information is ineffective due to: 1) it is impossible to compute the similarity between users/items when data are extremely sparse, and 2) even if some similarity information is calculated, this information may be too limited to recommend correct items. So its prediction performance gets worse when rating

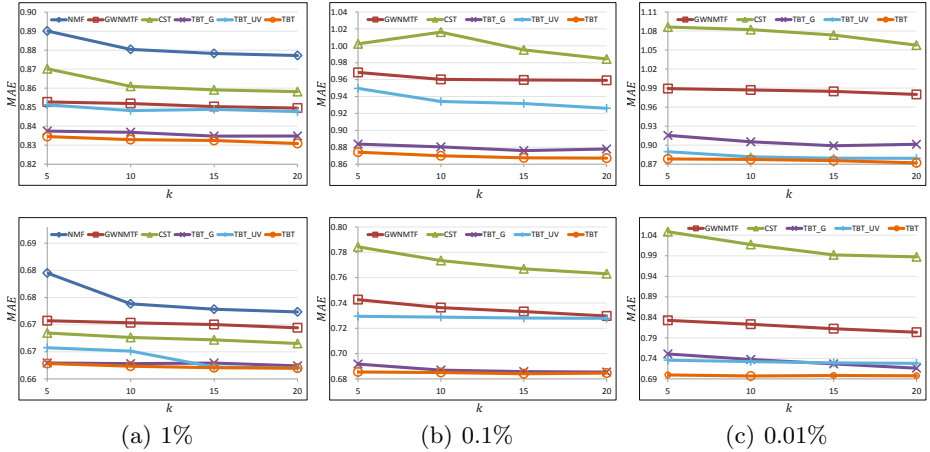


Fig. 3. Impact of #latent factors on Epinions (above) and MovieLens10M (bottom)

data are coming more sparse. To handle this problem, we construct a twin bridge, which can transfer more knowledge from dense auxiliary data to the target data.

Secondly, towards the negative transfer issue, CST encounters the problem that the transferred latent factors extracted from the auxiliary data may contain some negative information for the target data. Our twin bridge transfer mechanism alleviates this problem by using similarity graphs as regularization when training the recommendation model. Notably, the graph regularization can filter out the negative information introduced by the transferred latent factors. With this advantage, TBT is made more robust to the extremely sparse data.

- Although TBT is a combination of our proposed TBT_{UV} and TBT_G , it outperforms both of them. This proves that the twin bridge transfer can reinforce the learning of both bridges. Each bridge emphasizes different properties of the data and enriches the matrix tri-factorization with complementary information.

- Unfortunately, transfer learning method CST has not consistently outperformed non-transfer learning methods GWNMTF. CST takes advantage on moderately sparse data (0.1%~1%) by knowledge transfer. However, under extremely sparsity levels (0.01%~0.1%), the latent factors transferred by CST may contain some negative information for the target data. In this case, GWNMTF performs better than CST, since it is not affected by the negative transfer issue.

5.5 Parameter Sensitivity

We investigate the parameter sensitivity by varying the values of k , λ and γ under different sparsity levels. As Figure 3 shows, under a given sparsity level, the more latent factors used, the better the performance is. Typically, we set $k = 20$ as used by baseline methods. For λ and γ , We set $\lambda = \gamma$ in TBT for easy comparison with baselines, and report average MAE in Figure 4. Each column corresponds

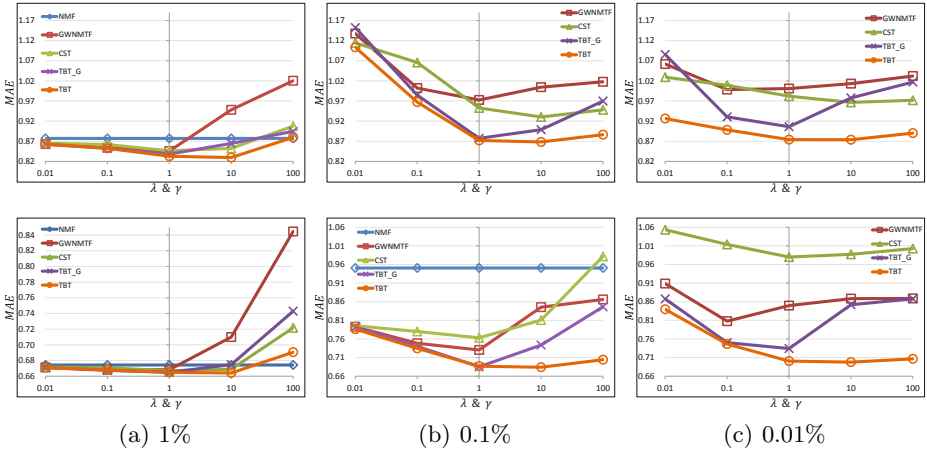


Fig. 4. Regularization param impact on Epinions (above)/MovieLens10M (bottom)

to a specific sparsity level of training ratings while each row corresponds to a different dataset. We observe that, 1) TBT method performs much more stably under different parameter values and sparsity levels than the baseline methods, and 2) neither non-transfer method GWNMTF or single bridge transfer methods CST/TBT_G can perform stably under different parameter values, sparsity levels, or data sets. Luckily, our TBT benefits from its twin bridge mechanism and performs much more robustly under all of these experimentation settings.

6 Conclusion

In this paper, we proposed a novel approach, Twin Bridge Transfer Learning (TBT), to reduce the sparsity in the target data. TBT consists of two sequential steps: 1) TBT extracts latent factors of users and items from dense auxiliary data and constructs similarity graphs from them; 2) TBT utilizes the latent factors and similarity graphs as twin bridge to transfer knowledge from dense auxiliary data to sparse target data. By using the twin bridge, TBT can enhance sufficient transfer by transferring more knowledge, while alleviate negative transfer by regularizing the learning model with the similarity graphs, which can naturally filter out the negative information contained in the latent factors. Experiments on real-world data sets validate the effectiveness of the proposed TBT approach.

Acknowledgement. The work is supported by the National HGJ Key Project (No. 2010ZX01042-002-002-01), the 973 Program of China (No. 2009CB320706), the National Natural Science Foundation of China (No. 61050010, No. 61073005, No. 60972096).

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *TKDE* 17 (2005)
2. Ma, H., King, I., Lyu, M.R.: Learning to recommend with social trust ensemble. In: *SIGIR* (2009)
3. Song, Y., Zhuang, Z., Li, H., Zhao, Q., Li, J., Lee, W.C., Giles, C.L.: Real-time automatic tag recommendation. In: *SIGIR* (2008)
4. Gu, Q., Zhou, J., Ding, C.: Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs. In: *SDM* (2010)
5. Basilico, J., Hofmann, T.: Unifying collaborative and content-based filtering. In: *ICML* (2004)
6. Melville, P., Mooney, R., Nagarajan, R.: Content-boosted collaborative filtering for improved recommendations. In: *AAAI* (2002)
7. Cao, B., Liu, N., Yang, Q.: Transfer learning for collective link prediction in multiple heterogenous domains. In: *ICML* (2010)
8. Li, B., Yang, Q., Xue, X.: Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. In: *AAAI* (2009)
9. Li, B., Yang, Q., Xue, X.: Transfer learning for collaborative filtering via a rating-matrix generative model. In: *ICML* (2009)
10. Pan, S.J., Yang, Q.: A survey on transfer learning. *TKDE* 22 (2010)
11. Pan, W., Liu, N.N., Xiang, E.W., Yang, Q.: Transfer learning to predict missing ratings via heterogeneous user feedbacks. *IJCAI* (2011)
12. Pan, W., Xiang, E., Liu, N., Yang, Q.: Transfer learning in collaborative filtering for sparsity reduction. In: *AAAI* (2010)
13. Pan, W., Xiang, E., Yang, Q.: Transfer learning in collaborative filtering with uncertain ratings. In: *AAAI* (2012)
14. Bell, R.M., Koren, Y.: Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In: *ICDM* (2007)
15. Long, M., Wang, J., Ding, G., Shen, D., Yang, Q.: Transfer learning with graph co-regularization. In: *AAAI* (2012)
16. Long, M., Wang, J., Ding, G., Cheng, W., Zhang, X., Wang, W.: Dual transfer learning. In: *SDM* (2012)
17. Herlocker, J., Konstan, J.A., Riedl, J.: An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval* 5 (2002)
18. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In: *NIPS*, vol. 14 (2001)
19. Boyd, S., Vandenberghe, L.: *Convex optimization*. Cambridge University Press (2004)
20. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: *NIPS* (2008)
21. Zhang, S., Wang, W., Ford, J., Makedon, F.: Learning from incomplete ratings using non-negative matrix factorization. *SIAM* (2006)