Probabilistic Matrix Factorization Based on Similarity Propagation and Trust Propagation for Recommendation

Haiyan Zhao
School of Optical-Electrical and Computer Engineering
University of Shanghai for Science and Technology
Shanghai, China
zhaohaiyan1992@foxmail.com

Shengsheng Wang
School of Optical-Electrical and Computer Engineering
University of Shanghai for Science and Technology
Shanghai, China
wangshengsheng6105@126.com

Qingkui Chen
School of Optical-Electrical and Computer Engineering
University of Shanghai for Science and Technology
Shanghai, China
chenqingkui@usst.edu.cn

Jian Cao
Department of Computer Science and Technology
Shanghai Jiao Tong University
Shanghai, China
cao-jian@cs.sjtu.edu.cn

***Abstract*—Traditional collaborative filtering approaches are often confronted with two major problems: data sparsity and cold-start. Fortunately, along with the rise of social media, social network is producing a large and rich set of social data (such as labels, trust, etc.), which provides a new way to solve the problems of collaborative filtering, namely, we can make use of social data to enhance the recommendation accuracy. However, traditional recommendation algorithms may only consider either the influence of similarity relationships or trust relationships to the user model, but fail to take full advantage of the implications of social data. In this paper, we propose a novel recommendation algorithm called STPMF based on neighborhood model and matrix factorization model, where complementary roles of similarity relationships and trust relationships to the user model by means of a weight w are considered simultaneously. Furthermore, we propagate similarity relationships and trust relationships one step or two steps to alleviate the data sparsity and cold-start problems. We have conducted experiments on two real world data sets from Last.fm and Delicious. Compared with existing recommendation algorithms, our method can effectively alleviate the problems of collaborative filtering, and enhance the recommendation accuracy.**

Keywords-collaborative filtering; social network; neighborhood model; matrix factorization model; similarity propagation; trust propagation

I. INTRODUCTION

Currently, recommender system has become more and more important and popular, which has widely applied in electronic commerce, social networks and web search, such as Amazon [1], Netflix [2], TiVo [3], etc. They also enable us to mine users' historical records and recommend items(such as movies, books, music, etc.) to users based on the preferences other users have expressed for those items.

Generally two types of Collaborative Filtering have been investigated: neighborhood model (also known as K Nearest Neighbor, or KNN) [1][4][5] and matrix factorization model [6][7]. Neighborhood model (KNN) can be further classified into two categories: user-oriented KNN method (UKNN) and item-oriented KNN method (IKNN). In reality, neighborhood methods are effective for capturing the local data information, while matrix factorization models are generally effective for capturing the global data information [8][9]. Therefore, we can incorporate neighborhood information into the factorization of the rating matrix, and effectively combine the advantages of neighborhood model and matrix factorization model.

However, due to the nature of collaborative filtering, recommender systems based on this technique still suffer from data sparsity and cold-start. In [5], the density of the available user-item rating matrix in commercial recommender systems is often less than 1%. Fortunately, along with the rise of social media (such as Facebook, Twitter, etc.), social network is producing a large and rich set of social data, which provides a new way to solve the problems of collaborative filtering, namely, we can make full use of social data (such as labels, trust, etc.) to enhance the recommendation accuracy.

Tags offer users an alternative way to represent their opinions about items. Since tags are originally created to organize resources in a user’s own way, they contain meaningful concepts to users. Therefore, they serve as a bridge representing semantic relationships between users and items [10][11]. In [12][13], it has shown that social tags can improve the performance of recommender systems. In [14][15], it is proved when the available rating matrix is extreme sparsity, it is hard for rating based KNNs to find the reliable nearest neighbors, since they assume users have at least rated some items in common (UKNN) or two items have been co-rated by some users (IKNN). Therefore, it would be a very significant attempt to use social tags to find local information for collaborative filtering in order to alleviate the data sparsity problems. In [14], similarity relationships are propagated to alleviate the data sparsity and cold-start problems in order to further enhance the recommendation accuracy. However, it doesn't consider the influence of trust relationships to the user model.

With the rapid development of social networks, social recommendation based on social networks has emerged, social trust can improve the performance of recommender systems [16][17]. In [18][19], social recommendation can

978-1-5090-0089-0/15 $31.00 © 2015 IEEE
DOI 10.1109/CIC.2015.11

alleviate the cold-start problems, since cold-start users are more dependent on the social network comparing to users with more ratings. For a user with very few ratings, it is hard to find similar users through similarity measures. However, if a user has quite a lot trust relationships, we can recommend items to him/her by means of trust relationships. In [18][20], trust relationships are propagated to alleviate the data sparsity and cold-start problems in order to further enhance the recommendation accuracy. Unfortunately, it doesn't consider the influence of similarity relationships to the user model.

Suppose Tom and Jack have very similar preferences, if Tom enjoys the film Avatar, then we can guess that Jack may also be interested in the film. Assuming that Tom and David are best friends, if Tom recommends the film Avatar to David, then David may have high probability to watch the film, because you often tend to trust your friends recommendations. However, the similarity relationships don't necessarily mean that two persons have trust relationships, the trust relationships don't necessarily mean that similarity relationships exist as well. In general, users' behaviors are influenced by similar users and the friends they trust. From the point of this view, similarity relationships and trust relationships play complementary roles to users' behaviors.

In this paper, in order to take full advantage of social tags and social trust data information to alleviate the data sparsity and cold-start problems, we propose a novel recommendation algorithm called STPMF based on neighborhood model and matrix factorization model, where complementary functions of similarity relationships and trust relationships to the user model by means of a weight w are considered simultaneously. Furthermore, we propagate similarity relationships and trust relationships in the network.

The contributions of this paper are as follows:

- We effectively combine the advantages of neighborhood model and matrix factorization model to capture more information from the data.
- We simultaneously consider complementary functions of similarity relationships and trust relationships to the user model through a weight w.
- We propagate similarity relationships and trust relationships to alleviate the data sparsity and cold-start problems.

The rest of this paper is organized as follows: We discuss some related work in Section 2. Probabilistic matrix factorization model is presented in Section 3. We introduce our proposed model in Section 4. The implementation of our proposed model in detail in Section 5. Our experiments are reported in Section 6. Finally, we conclude the paper and present some directions for future work.

II. RELATED WORK

Although neighborhood model and matrix factorization model have been widely studied in collaborative filtering respectively, only several researches have focused on combining these two models to enhance the recommendation accuracy [8][9][21]. In [8], researchers combine neighborhood model and matrix factorization model, the latent features of users and items learn from rating matrix through stochastic gradient descent (SGD). In [9], during the Netflix competition, researchers have found that there are complementary advantages for local neighborhood model and global matrix factorization model, and combining these two models could enhance the recommendation accuracy. In [14], NHPMF model incorporates user-oriented neighborhood model and item-oriented neighborhood model into the factorization of the rating matrix to ensure similar users (items) will have similar latent features. In [18], SocialMF model combines user-oriented neighborhood model and matrix factorization model, and the latent features of users are learned from the trust relationships. Our STPMF model effectively combines the advantages of neighborhood model and matrix factorization model, and captures more information from the data.

With the development of Web 2.0, tags have become a popular tool for users to describe items. In [10][11], researchers indicate tags may well represent users' interests and semantics of the items. Recently, many scholars have combined social tags and matrix factorization model to enhance the recommendation accuracy [14][22][23]. In [14], NHPMF model calculates user-oriented KNN and item-oriented KNN by means of tags, then fuses neighborhood information into the matrix factorization model, and simultaneously propagates similarity relationships of users and items to enhance the recommendation accuracy. In [22], TagRec model builds a unified probabilistic matrix factorization by utilizing users' tagging information and rating information. In [23], SoRecUser model connects users’ rating information with users’ tagging information through a shared user latent feature space. SoRecItem model connects items’ received rating information with items’ received tagging information through a shared item latent feature space. In [22][23], researchers don't take into account similarity propagation, NHPMF model [14] takes into account similarity propagation, but only considers the role of similarity relationships to the user model. Our STPMF model considers complementary roles of similarity relationships and trust relationships to the user model by means of a weight w.

With the rapid development of social networks, an increasing amount of data is becoming available on the internet, and social recommendation based on social networks has emerged [18][24][25]. In [18], SocialMF model forces the user feature vectors to be close to those of their friends in order to learn the latent features for users with no or very few ratings, and simultaneously propagates similarity relationships to enhance the recommendation accuracy. In [24], SoRec model connects users' rating information with users' social information through a shared user latent feature space. In [25], RSTE model assumes every user has his/her own taste and at the same time, every user may be influenced by his/her friends he/she trusts. In [24][25], researchers don't take into account trust propagation, SocialMF model [18] takes into account trust propagation, but only considers the role of trust relationships to the user model. Our STPMF model considers complementary roles of similarity relationships and trust relationships to the user model.

In summary, we propose a novel recommendation algorithm called STPMF, where complementary roles of similarity relationships and trust relationships to the user model by means of a weight w are considered. Furthermore, we simultaneously propagate similarity relationships and trust relationships to alleviate the data sparsity and cold-start problems to further enhance the recommendation accuracy. We have conducted experiments on two real world data sets from Last.fm and Delicious and achieved good results.

III. PROBABILISTIC MATRIX FACTORIZATION

Recommendation algorithms based on probabilistic matrix factorization predict user ratings for items from the perspective of probabilistic approaches. Assuming the observed ratings obey Gauss distribution, and also zero mean Gauss priors are assumed for user and item latent feature vectors. We can obtain the posterior probability of the latent variables U and V through a Bayesian inference, and then maximize the posterior probability. In general, we obtain the log of the posterior probability firstly, and then take derivatives of the latent variables U and V by means of stochastic gradient descent (SGD). Table I lists the mathematical notations used in this paper.

TABLE I. MATHEMATICAL NOTATIONS

Notation	Description
M, N, L	The number of users, items, tags respectively
F	Dimension of latent feature representation
R_{M*N}	User-item rating matrix
U_{M*F}	User's latent feature matrix
V_{N*F}	Item's latent feature matrix
N_{U_i}	The top K set of similar neighbors of user U_i
N_{V_j}	The top K set of similar neighbors of item V_j
T_{U_i}	The top K set of trust neighbors of user U_i
$\hat{R}_{ij}$	Approximation of the rating matrix
$\sigma_R^2, \sigma_U^2, \sigma_V^2$	The variance of rating, user and item latent feature vectors respectively
$\sigma_{ST}^2, \sigma_S^2$	The variance of user and item model respectively

In order to learn the latent features of users and items, we employ matrix factorization to factorize the user-item rating matrix. The conditional probability of the observed ratings is defined as:

$$p(R|U,V,\sigma_R^2) = \prod_{i=1}^{M}\prod_{j=1}^{N}\left[N\left(R_{ij}|U_i^T V_j,\sigma_R^2\right)\right]^{I_{ij}^R} \quad (1)$$

Where $N(x|\mu,\sigma^2)$ is the normal distribution with mean μ and variance σ^2. I_{ij}^R is the indicator function that is equal to 1 if U_i has rated V_j and equal to 0 otherwise. Also, zero mean Gauss priors are assumed for user and item latent feature vectors:

$$p(U|\sigma_U^2) = \prod_{i=1}^{M} N(U_i|0,\sigma_U^2\mathbf{I}) \quad (2)$$

$$p(V|\sigma_V^2) = \prod_{j=1}^{N} N\left(V_j|0,\sigma_V^2\mathbf{I}\right) \quad (3)$$

Now, through a Bayesian inference, the posterior probability of the latent variables U and V can be obtained as follows:

$$\begin{aligned} p(U,V|R,\sigma_R^2,\sigma_U^2,\sigma_V^2) &\propto p(R|U,V,\sigma_R^2)p(U|\sigma_U^2)p(V|\sigma_V^2) \\ &= \prod_{i=1}^{M}\prod_{j=1}^{N}\left[N\left(R_{ij}|U_i^T V_j,\sigma_R^2\right)\right]^{I_{ij}^R} \times \prod_{i=1}^{M} N(U_i|0,\sigma_U^2\mathbf{I}) \\ &\times \prod_{j=1}^{N} N\left(V_j|0,\sigma_V^2\mathbf{I}\right) \end{aligned} \quad (4)$$

The corresponding graphical model is presented in Fig. 1. Using (4), we can learn the latent feature vectors of users and items purely based on the user-item rating matrix.

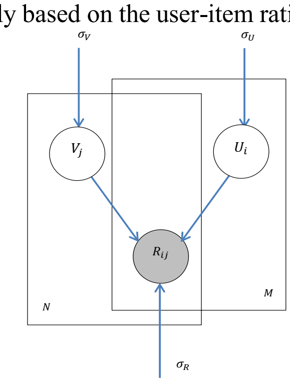

Figure 1. Graphical model for probabilistic matrix factorization

IV. THE PROCESS OF BUILDING STPMF

A. The Process of Building Similarity Network

We define two matrices by means of co-occurrence relationships: UT (user-tag matrix), VT (item-tag matrix), the element UT_{ik} of UT represents the number of items that are annotated with the tag T_k by the user U_i, the element VT_{jk} of VT represents the number of users that annotate the item V_j with the tag T_k.

The similarity can be measured by computing the cosine similarity between two users or items:

$$\begin{cases} sim(U_i,U_j) = cos(U_i,U_j) \\ sim(V_i,V_j) = cos(V_i,V_j) \\ cos\left(\vec{i},\vec{j}\right) = \dfrac{\vec{i}\cdot\vec{j}}{\left\|\vec{i}\right\|_F \left\|\vec{j}\right\|_F} \end{cases} \quad (5)$$

The edge $sim(vk,j)$ represents the similarity between the item V_j and its similar neighbor V_{vk}, the edge $sim(i,uk)$ represents between the similarity of the user U_i and his/her similar neighbor U_{uk}. The corresponding graphical model based on similarity network is presented in Fig. 2.

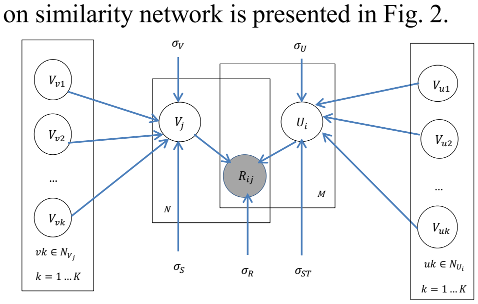

Figure 2. Probabilistic graphical model NHPMF[14] based on similarity network

B. The Process of Building Trust Network

In this paper, the trust network belongs to undirected multivalued network, where $N(U)$ is the friends set that user U trusts, and $|N(U)|$ is the number of trusted friends of user U in the set $N(U)$. The trust degree can be calculated on undirected multivalued trust network as follows:

$$trust(U_1, U_2) = \frac{|N(U_1) \cap N(U_2)|}{|N(U_1) \cup N(U_2)|} \tag{6}$$

The edge $tru(i, nk)$ represents the trust degree between the user U_i and his/her trust neighbor U_{nk}. The corresponding graphical model based on trust network is presented in Fig. 3.

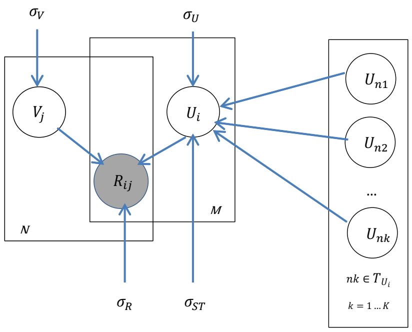

Figure 3. Probabilistic graphical model based on trust network

C. The Framework of STPMF Model

In this section, we present our approach of incorporating user similarity network, item similarity network and user trust network into a matrix factorization model for recommendation. The framework of STPMF model can be divided into four-stage as follows:

1) to calculate user similar K-Nearest Neighbor through user similarity network, item similar K-Nearest Neighbor through item similarity network, and user trust K-Nearest Neighbor through user trust network respectively.

2) to consider complementary roles of similarity relationships and trust relationships to the user model by means of a weight w, and similarity relationships to the item model simultaneously.

3) to incorporate neighborhood information into matrix factorization model, and then iterative learning from the training data set by means of stochastic gradient descent (SGD).

4) to obtain the final latent variables U and V, and then according to the formula $\hat{R}_{ij} = U_i^T V_j$ to predict the preference of user U_i for the item V_j.

The edge $sim(vk, j)$ represents the similarity between the item V_j and its similar neighbor V_{vk}, the edge $sim(i, uk)$ represents between the similarity of the user U_i and his/her similar neighbor U_{uk}. The edge $tru(i, nk)$ represents the trust degree between the user U_i and his/her trust neighbor U_{nk}. We simultaneously consider complementary roles of similarity relationships and trust relationships to the user model through a weight w. The corresponding graphical model is presented in Fig. 4.

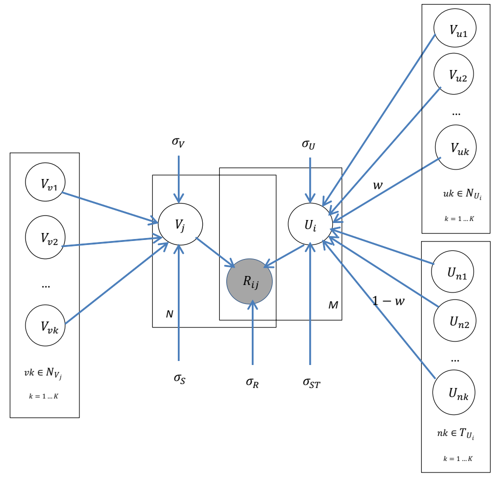

Figure 4. Probabilistic graphical model of STPMF

V. THE IMPLEMENTATION OF STPMF

A. The Process of Deriving STPMF

Intuitively, the behavior of user U_i is similar to that his/her similar neighbor set N_{U_i}, and the attributes of item V_j are similar to those of its similar neighbor set N_{V_j}, simultaneously the behavior of user U_i is affected by his/her trust neighbor set T_{U_i} as well. We consider complementary roles of similarity relationships and trust relationships to the user model by means of a weight w. Based on this intuition, we formulate the following equations:

$$\begin{cases} U_i = (w \sum_{m \in N_{U_i}} sim(i, m) \times U_m + (1 - w) \sum_{n \in T_{U_i}} tru(i, n) \times U_n) + \theta_U & (7) \\ V_j = \sum_{m \in N_{V_j}} sim(j, m) \times V_m + \theta_V & (8) \end{cases}$$

Where $\theta_U \sim N(0, \sigma_{ST}^2 \mathbf{I})$, $\theta_V \sim N(0, \sigma_S^2 \mathbf{I})$. In the process of calculation, we normalize the similarity and trust degree to ensure $\sum_{m \in N_{U_i}} sim(i, m) = 1$, $\sum_{n \in T_{U_i}} tru(i, n) = 1$ and $\sum_{m \in N_{V_j}} sim(j, m) = 1$. In the above two equations, each user's and item's latent feature vector is consisted of two terms. The first term denotes the group feature of the user (item), which is the weighted average of his/her (its) neighborhood. The second term underlines the uniqueness of each user and item feature vector, which is different from from his/her (its) neighborhood to an extent. The divergence is controlled by the variance parameter σ_{ST}^2 and σ_S^2 in (7) and (8). The smaller the variance, the less possible that the feature vector is different from that of his/her (its) neighbor set.

For the user latent features, we have two factors: the zero mean Gaussian prior to avoid over-fitting, and the conditional distribution of the user latent features given the latent features of his/her similar neighbors and trust neighbors. Therefore, the user model is shown below:

$$\begin{aligned} p(U|S, T, \sigma_U^2, \sigma_{ST}^2) &\propto p(U|\sigma_U^2) \times p(U|S, T, \sigma_{ST}^2) \\ &= \prod_{i=1}^{M} N(U_i|0, \sigma_U^2 \mathbf{I}) \\ &\times \prod_{i=1}^{M} N\Bigg(U_i | w \sum_{m \in N_{U_i}} sim(i, m) \times U_m \\ &+ (1 - w) \sum_{n \in T_{U_i}} tru(i, n) \times U_n, \sigma_{ST}^2 \mathbf{I}\Bigg) \end{aligned} \tag{9}$$

For the item latent features, we have two factors: The zero mean Gaussian prior to avoid over-fitting, and the conditional distribution of item latent features given the latent features of its similar neighbors. Therefore, the item model is shown below:

$$p(V|S,\sigma_V^2,\sigma_S^2) \propto p(V|\sigma_V^2) \times p(V|S,\sigma_S^2) = \prod_{j=1}^{M} N(V_j|0,\sigma_V^2\mathbf{I}) \times \prod_{j=1}^{N} N\left(V_j | \sum_{m\in N_{V_j}} sim(j,m) \times V_m, \sigma_S^2\mathbf{I}\right) \quad (10)$$

Note that taking the neighborhood information into account doesn't change the conditional distributions of the observed ratings. It only affects the user and item latent feature vectors. Therefore, the conditional distribution over the observed rating is:

$$p(R|U,V,\sigma_R^2) = \prod_{i=1}^{M}\prod_{j=1}^{N}\left[N(R_{ij}|U_i^T V_j,\sigma_R^2)\right]^{I_{ij}^R} \quad (11)$$

According to (9), (10) and (11), through a Bayesian inference, the posterior distribution over user and item latent factors is given by (12):

$$\begin{aligned} p(U,V|R,S,T,\sigma_R^2,\sigma_{ST}^2,\sigma_S^2,\sigma_U^2,\sigma_V^2) &\propto p(R|U,V,\sigma_R^2) \times p(U|S,T,\sigma_U^2,\sigma_{ST}^2) \times p(V|S,\sigma_V^2,\sigma_S^2) \\ &= \prod_{i=1}^{M}\prod_{j=1}^{N}\left[N(R_{ij}|U_i^T V_j,\sigma_R^2)\right]^{I_{ij}^R} \times \prod_{i=1}^{M} N(U_i|0,\sigma_U^2\mathbf{I}) \\ &\times \prod_{i=1}^{M} N\left(U_i | w\sum_{m\in N_{U_i}} sim(i,m)\times U_m + (1-w)\sum_{n\in T_{U_i}} tru(i,n)\times U_n, \sigma_{ST}^2\mathbf{I}\right) \\ &\times \prod_{j=1}^{N} N(V_j|0,\sigma_V^2\mathbf{I}) \\ &\times \prod_{j=1}^{N} N\left(V_j | \sum_{m\in N_{V_j}} sim(j,m)\times V_m, \sigma_S^2\mathbf{I}\right) \end{aligned} \quad (12)$$

Keeping the hyperparameters $(\sigma_R^2, \sigma_{ST}^2, \sigma_S^2, \sigma_U^2, \sigma_V^2)$ fixed, maximizing the log posterior in (12) is equivalent to minimizing the following objective function:

$$\begin{aligned} \mathcal{L}(U,V,R,S,T) &= \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{N} I_{ij}^R (R_{ij} - U_i^T V_j)^2 + \frac{\lambda_U}{2}\sum_{i=1}^{M}\|U_i\|_F^2 \\ &+ \frac{\lambda_V}{2}\sum_{j=1}^{N}\|V_j\|_F^2 + \frac{\lambda_{ST}}{2}\sum_{i=1}^{M}\left\| U_i - \left(w\sum_{m\in N_{U_i}} sim(i,m)\times U_m + (1-w)\sum_{n\in T_{U_i}} tru(i,n)\times U_n\right)\right\|_F^2 \\ &+ \frac{\lambda_S}{2}\sum_{j=1}^{N}\left\| V_j - \sum_{m\in N_{V_j}} sim(j,m)\times V_m\right\|_F^2 \end{aligned} \quad (13)$$

Where $\lambda_U = \sigma_R^2/\sigma_U^2$, $\lambda_V = \sigma_R^2/\sigma_V^2$, $\lambda_{ST} = \sigma_R^2/\sigma_{ST}^2$, $\lambda_S = \sigma_R^2/\sigma_S^2$ and $\|\cdot\|_F^2$ denotes the Frobenius norm.

We can find a local minimum of the objective function in (13) by performing stochastic gradient descent (SGD) on U_i and V_j as follows:

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial U_i} &= \sum_{j=1}^{N} I_{ij}^R (-V_j)(R_{ij} - U_i^T V_j) + \lambda_U U_i \\ &+ \lambda_{ST}\left(U_i - \left(w\sum_{m\in N_{U_i}} sim(i,m)\times U_m + (1-w)\sum_{n\in T_{U_i}} tru(i,n)\times U_n\right)\right) \\ &- w\lambda_{ST}\sum_{\{m|i\in N_{U_m}\}} sim(m,i)\left(U_m - \sum_{w\in N_{U_m}} sim(w,m)\times U_w\right) \\ &- (1-w)\lambda_{ST}\sum_{\{n|i\in T_{U_n}\}} tru(n,i)\left(U_n - \sum_{w\in T_{U_n}} tru(w,n)\times U_w\right) \end{aligned} \quad (14)$$

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial V_j} &= \sum_{i=1}^{M} I_{ij}^R (-U_i)(R_{ij} - U_i^T V_j) + \lambda_V V_j + \lambda_S\left(V_j - \sum_{m\in N_{V_j}} sim(j,m)\times V_m\right) \\ &- \lambda_S\sum_{\{m|j\in N_{V_m}\}} sim(m,j)\left(V_m - \sum_{w\in N_{V_m}} sim(w,m)\times V_w\right) \end{aligned} \quad (15)$$

In order to reduce the model complexity, we set $\lambda_U = \lambda_V$, $\lambda_{ST} = \lambda_S$. In each iteration, U and V are updated based on the latent variables from the previous iteration.

B. Discussion on STPMF

Naturally, the behavior of user U_i is similar to that his/her similar neighbor set N_{U_i}, and the attributes of item V_j are similar to those of its similar neighbor set N_{V_j}, simultaneously the behavior of user U_i is affected by his/her trust neighbor set T_{U_i} as well. Based on the above intuition, we consider complementary roles of similarity relationships and trust relationships to the user model by means of a weight w, and similarity relationships to the item model. Therefore, STPMF model is different from related PMF model, where it assumes all users and items obey the same zero mean Gaussian distribution, which is too rough to capture user's and item's personality.

We call this model TPMF, if only consider trust relationships to the user model, and similarity relationships to the item model. To get deeper insights, we observe that our STPMF model is actually a unified framework of KNN, PMF, NHPMF and TPMF. Under certain circumstances, it degenerates to the following methods:

1) when $w = 0$, STPMF model degenerates to TPMF model.

a) $\forall i,j \in userset: tru(i,j) = 0$, and $\forall i,j \in itemset: sim(i,j) = 0$: In this case, we have no other external data sources to extract the neighbors of users and items. Thus, users and items' feature vectors turn to be the zero mean Gaussian prior. The model degenerates to PMF model.

b) $\forall i,j \in userset: tru(i,j) = 0$: In this case, we have no external data sources to extract the neighbors of users, but we calculate the similarities between items. Therefore, user feature vectors obey a common zero mean Gaussian distribution. While the feature vector of each item still has a unique Gaussian distribution with its mean centered around its similar neighbors. The model turns to the fusion of IKNN and PMF. This scenario is common in e-commerce sites, such as Amazon, Taobao, etc.

c) $\forall i,j \in itemset: sim(i,j) = 0$: In this case, we have no external data sources to extract the neighbors of items, but we calculate the trust degree between users. Therefore, item feature vectors obey a common zero mean Gaussian distribution. While the feature vector of each user still has a unique Gaussian distribution with his/her mean centered around his/her trust neighbors. The model turns to the fusion of UKNN and PMF. This scenario is common in the social networks, such as Twitter, Epinions, etc.

2) when $w = 1$, STPMF model degenerates to NHPMF model.

a) $\forall i,j \in userset: sim(i,j) = 0$, and $\forall i,j \in itemset: sim(i,j) = 0$.

b) $\forall i,j \in userset: sim(i,j) = 0$.

c) $\forall i,j \in itemset: sim(i,j) = 0$.

Under this situation, the analysis is also similar to the situation above, and we don't give out the concrete analysis of this situation any longer.

VI. EXPERIMENT AND ANALYSIS

We conduct several experiments to compare our models with existing methods. Our experiments are mainly focused on the following questions:

1. How does our approach perform comparing with the existing state-of-the-art collaborative filtering algorithms?
2. How does the model parameter λ_2 affect the accuracy of recommendation?
3. How does the model parameter w affect the accuracy of recommendation?
4. Can our algorithm alleviate the problems of collaborative filtering effectively?
5. How does the propagation mechanism affect the accuracy of recommendation?

A. Datasets Description

All experiments are performed on two real world datasets: Last.fm and Delicious. More statistics of the datasets can be seen in Table II.

TABLE II. DATASETS DESCRIPTION

Statistics	Last.fm	Delicious
# of users	1,892	1,737
# of items	17,632	5,633
# of tags	11,946	14,192
# of rating records	92,834	2,2767
# of friend records	12,717	7,668
Average tags per user	98.562	58.482
Average tags per item	14.891	6.321

For Last.fm dataset, if the user has listened to an artist (item) then we consider the user rating for the artist as 1, and otherwise 0. Similarity, for Delicious dataset, if the user has bookmarked an URL (item) then we consider the rating for that bookmarked URL as 1, and otherwise 0. We consider artists and URLs as items in the above two datasets. The latter dataset is first preprocessed to remove the items that are bookmarked less than or equal to two times. We observe that the user-item matrices for the datasets are highly sparse (99.7217% sparse and 99.7673% sparse respectively).

B. Experimental Setup

We perform 5-fold cross validation in our experiments. In each fold, we split each of the datasets into two parts – training (80%) and testing datasets (20%). The evaluation metrics we use are mean absolute error (MAE) and root mean square error (RMSE), which is defined as:

$$MAE = \frac{\sum_{ij}|R_{ij} - \hat{R}_{ij}|}{N} \tag{16}$$

$$RMSE = \sqrt{\frac{\sum_{ij}\left(R_{ij} - \hat{R}_{ij}\right)^2}{N}} \tag{17}$$

Where R_{ij} denotes the real rating user U_i give to item V_j, $\hat{R}_{ij}$ denotes the predicted rating user U_i give to item V_j, and N is the number of tested ratings. The smaller the MAE and RMSE values, the higher the recommendation accuracy.

C. Experimental Results

1) Last.fm dataset

We set $\lambda_U = \lambda_V = \lambda_1 = 0.05$, $\lambda_S = \lambda_{ST} = \lambda_2 = 0.01$, and otherwise learning rate $\alpha = 0.02$, $w = 0.5$. The experimental results as follows:

TABLE III. MAE COMPARISONS WITH OTHER APPROACHES

	UKNN	IKNN	PMF	SocialMF	NHPMF	STPMF
K=5, D=5	0.8926	0.8913	0.6986	0.6473	0.6803	0.6385
K=5, D=10	0.8926	0.8913	0.6765	0.6582	0.6508	0.6142
K=5, D=15	0.8926	0.8913	0.6392	0.6161	0.6199	0.5870
K=5, D=20	0.8926	0.8913	0.6157	0.6139	0.6035	0.6018
K=10, D=5	0.8356	0.8467	0.6986	0.6473	0.6420	0.6409
K=10, D=10	0.8356	0.8467	0.6765	0.6582	0.6220	0.6197
K=10, D=15	0.8356	0.8467	0.6392	0.6161	0.6147	0.5804
K=10, D=20	0.8356	0.8467	0.6157	0.6139	0.6035	0.5780
K=15, D=5	0.8021	0.8295	0.6986	0.6473	0.6408	0.6168
K=15, D=10	0.8021	0.8295	0.6765	0.6582	0.6474	0.6136
K=15, D=15	0.8021	0.8295	0.6392	0.6161	0.6204	0.5930
K=15, D=20	0.8021	0.8295	0.6157	0.6139	0.6045	0.5840
K=20, D=5	0.7820	0.8293	0.6986	0.6473	0.6518	0.6404
K=20, D=10	0.7820	0.8293	0.6765	0.6582	0.6441	0.6228
K=20, D=15	0.7820	0.8293	0.6392	0.6161	0.6216	0.5874
K=20, D=20	0.7820	0.8293	0.6157	0.6139	0.6048	0.5913

TABLE IV. RMSE COMPARISONS WITH OTHER APPROACHES

	UKNN	IKNN	PMF	SocialMF	NHPMF	STPMF
K=5, D=5	0.9124	0.9218	0.7555	0.7173	0.7417	0.7097
K=5, D=10	0.9124	0.9218	0.7392	0.7255	0.7196	0.6918
K=5, D=15	0.9124	0.9218	0.7101	0.6923	0.6963	0.6693
K=5, D=20	0.9124	0.9218	0.6941	0.6924	0.6826	0.6807
K=10, D=5	0.8693	0.8996	0.7555	0.7173	0.7393	0.7168
K=10, D=10	0.8693	0.8996	0.7392	0.7255	0.6953	0.6902
K=10, D=15	0.8693	0.8996	0.7101	0.6923	0.6910	0.6642
K=10, D=20	0.8693	0.8996	0.6941	0.6924	0.6823	0.6624
K=15, D=5	0.8448	0.9076	0.7555	0.7173	0.7110	0.6927
K=15, D=10	0.8448	0.9076	0.7392	0.7255	0.7171	0.6901
K=15, D=15	0.8448	0.9076	0.7101	0.6923	0.6908	0.6742
K=15, D=20	0.8448	0.9076	0.6941	0.6924	0.6835	0.6670
K=20, D=5	0.8320	0.9347	0.7555	0.7173	0.7196	0.7114
K=20, D=10	0.8320	0.9347	0.7392	0.7255	0.7133	0.6973
K=20, D=15	0.8320	0.9347	0.7101	0.6923	0.6952	0.6698
K=20, D=20	0.8320	0.9347	0.6941	0.6924	0.6840	0.6726

2) Delicious dataset

We set $\lambda_U = \lambda_V = \lambda_1 = 0.05$, $\lambda_S = \lambda_{ST} = \lambda_2 = 0.02$, and otherwise learning rate $\alpha = 0.001$, $w = 0.5$. The experimental results as follows:

TABLE V. MAE COMPARISONS WITH OTHER APPROACHES

	UKNN	IKNN	PMF	SocialMF	NHPMF	STPMF
K=5, D=5	0.8449	0.9226	0.7477	0.7314	0.7325	0.7276
K=5, D=10	0.8449	0.9226	0.7473	0.7279	0.7293	0.7250
K=5, D=15	0.8449	0.9226	0.7467	0.7275	0.7282	0.7240
K=5, D=20	0.8449	0.9226	0.7456	0.7272	0.7281	0.7227
K=10, D=5	0.8400	0.8995	0.7477	0.7314	0.7315	0.7255
K=10, D=10	0.8400	0.8995	0.7473	0.7279	0.7288	0.7236
K=10, D=15	0.8400	0.8995	0.7467	0.7275	0.7275	0.7234
K=10, D=20	0.8400	0.8995	0.7456	0.7272	0.7274	0.7226
K=15, D=5	0.8367	0.8816	0.7477	0.7314	0.7311	0.7251
K=15, D=10	0.8367	0.8816	0.7473	0.7279	0.7286	0.7245
K=15, D=15	0.8367	0.8816	0.7467	0.7275	0.7280	0.7244
K=15, D=20	0.8367	0.8816	0.7456	0.7272	0.7278	0.7236
K=20, D=5	0.8342	0.8699	0.7477	0.7314	0.7311	0.7269
K=20, D=10	0.8342	0.8699	0.7473	0.7279	0.7305	0.7254
K=20, D=15	0.8342	0.8699	0.7467	0.7275	0.7286	0.7245
K=20, D=20	0.8342	0.8699	0.7456	0.7272	0.7280	0.7239

TABLE VI. RMSE COMPARISONS WITH OTHER APPROACHES

	UKNN	IKNN	PMF	SocialMF	NHPMF	STPMF
K=5, D=5	0.8789	0.9446	0.7550	0.7398	0.7406	**0.7356**
K=5, D=10	0.8789	0.9446	0.7513	0.7334	0.7343	**0.7300**
K=5, D=15	0.8789	0.9446	0.7503	0.7314	0.7330	**0.7287**
K=5, D=20	0.8789	0.9446	0.7483	0.7305	0.7322	**0.7249**
K=10, D=5	0.8754	0.9230	0.7550	0.7398	0.7400	**0.7338**
K=10, D=10	0.8754	0.9230	0.7513	0.7334	0.7336	**0.7286**
K=10, D=15	0.8754	0.9230	0.7503	0.7314	0.7311	**0.7283**
K=10, D=20	0.8754	0.9230	0.7483	0.7305	0.7307	**0.7246**
K=15, D=5	0.8727	0.9057	0.7550	0.7398	0.7391	**0.7337**
K=15, D=10	0.8727	0.9057	0.7513	0.7334	0.7333	**0.7299**
K=15, D=15	0.8727	0.9057	0.7503	0.7314	0.7317	**0.7280**
K=15, D=20	0.8727	0.9057	0.7483	0.7305	0.7308	**0.7239**
K=20, D=5	0.8704	0.8943	0.7550	0.7398	0.7391	**0.7350**
K=20, D=10	0.8704	0.8943	0.7513	0.7334	0.7365	**0.7305**
K=20, D=15	0.8704	0.8943	0.7503	0.7314	0.7321	**0.7283**
K=20, D=20	0.8704	0.8943	0.7483	0.7305	0.7309	**0.7242**

Table 3 and Table 5 report the MAE values of all the algorithms under different settings of neighbor set size K and latent feature dimension D respectively. Table 4 and Table 6 report the RMSE values of all the algorithms under different settings of neighbor set size K and latent feature dimension D respectively. The neighbor set size K in neighborhood methods is set to 5, 10, 15, and 20 respectively while the dimensionality of the latent feature vectors D in matrix factorization methods is set to 5, 10, 15, and 20 respectively.

From the comparison, we have the following observations:

- Our method STPMF, which takes full advantage of social data, effectively combines the advantages of neighborhood model and matrix factorization model, captures more information from the data, and performs the best in all situations. Let's take $K = 10$, $D = 20$ as an example, for Last.fm dataset, compared with PMF, SocialMF and NHPMF, STPMF improves the MAE of the results by about 3.77%, 3.59% and 2.55%, and the RMSE of the results by about 3.17%, 3%, 1.99%. Under the same condition, for Delicious dataset, compared with PMF, SocialMF and NHPMF, STPMF improves the MAE of the results by about 2.3%, 0.46% and 0.48%, and the RMSE of the results by about 2.37%, 0.59% and 0.61%.
- In pure KNN methods, the increase of K in a certain range is applied to enhance the recommendation accuracy. However, there's no benefit to increase K in NHPMF and STPMF, because adding more neighbors will cover more global efforts.
- In general, larger values of D will give us more flexibility to represent users and items in the latent space, leading to better performance. However, with the increase of D, overfitting may be caused by complexity.
- Compared with PMF, SocialMPF, NHPMF and STPMF methods incorporate the mechanism of trust relationships or similarity relationships to improve the quality of recommendation. Therefore, making full use of social data can make better recommendations.

D. Impact of λ_2 on the Results

1) Last.fm dataset

We set $\alpha = 0.02$, $w = 0.5$, $\lambda_1 = 0.05$. The impact of λ_2 on the results as follows:

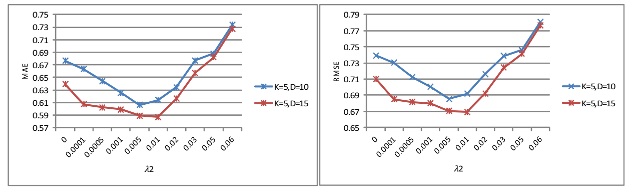

Figure 5. Impact of different values of λ_2 on the performance of prediction in Last.fm

2) Delicious dataset

We set $\alpha = 0.001$, $w = 0.5$, $\lambda_1 = 0.05$. The impact of λ_2 on the results as follows:

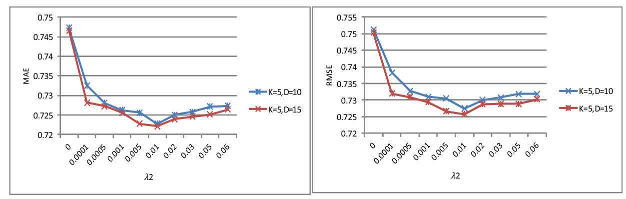

Figure 6. Impact of different values of λ_2 on the performance of prediction in Delicious

The parameter λ_2 controls the influence of the similar neighbors and trust neighbors on the behavior of users, and the influence of the similar neighbors on the attribute of items. Larger values of λ_2 in the objective function indicate more impact of the social data on the behavior of users and the attribute of items. Very small values of λ_2 makes our model close to the PMF model.

Fig. 5 and Fig. 6 compare the MAE and RMSE of our model for different ranges of values for λ_2 in both data sets. Let's take $K = 5$, $D = 10$ as an example, STPMF has its best results on Last.fm dataset for $\lambda_2 = 0.005$, and $\lambda_2 = 0.01$ for Delicious dataset. Let's take $K = 5$, $D = 15$ as an example, STPMF has its best results on $\lambda_2 = 0.01$ for Last.fm and Delicious dataset.

E. Impact of w on the Results

1) Last.fm dataset

We set $\alpha = 0.001$, $\lambda_1 = 0.05$, $\lambda_2 = 0.02$. The impact of w on the results as follows:

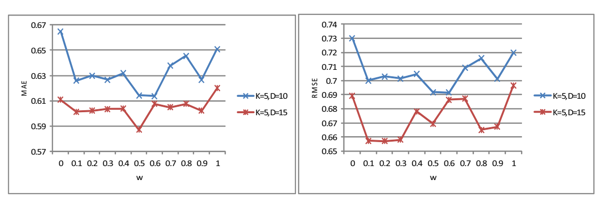

Figure 7. Impact of different values of w on the performance of prediction in Last.fm

2) Delicious dataset

We set $\alpha = 0.001$, $\lambda_1 = 0.05$, $\lambda_2 = 0.02$. The impact of w on the results as follows:

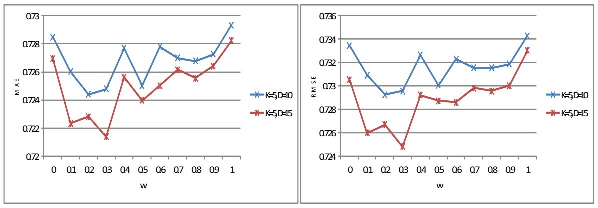

Figure 8. Impact of different values of w on the performance of prediction in Delicious

The parameter w balances the influence of the similar relationships and trust relationships on the behavior of users. If $w = 0$, we only consider trust relationships to the user model, namely, STPMF model degenerates to TPMF model. If $w = 1$, we only consider similarity relationships to the user model, namely, STPMF model degenerates to NHPMF model. If $w \in (0,1)$, we simultaneously consider complementary roles of similarity relationships and trust relationships to the user model by means of a weight w. As shown above in Fig. 7 and Fig. 8, when we simultaneously consider complementary roles of similarity relationships and trust relationships to the user model, and we get better results compared with only consider similarity relationships or trust relationships to the user model.

F. Performance on Different Users

We choose Last.fm dataset for this experiment. The parameter settings of our approach are $\alpha = 0.02$, $w = 0.5$, $\lambda_1 = 0.05$, $\lambda_2 = 0.01$, $K = 5$, $D = 15$. We summarize the user group distribution of the Last.fm dataset, and the experimental results as follows:

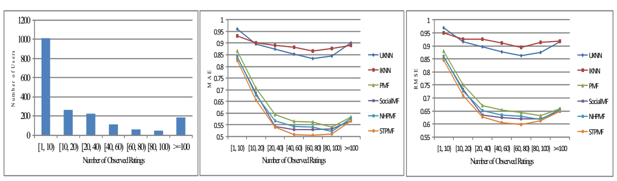

Figure 9. Performance on Different Group Users

Data sparsity is a major problem for collaborative filtering. Hence, in order to compare our approach with the other methods under this situation, we first group all the users based on the number of observed ratings in the Last.fm dataset, and then evaluate prediction accuracies of different user groups. The experimental results are shown in Fig. 9. Users are grouped into 7 classes: [1-10), [10-20), [20-40), [40-60), [60-80), [80-100) and ">=100", denoting how many ratings users have rated. Obviously, this is a very sparse dataset since 53.44% of the users in [1-10).

Pure CF methods, such as UKNN, IKNN and PMF, which only rely on the rating matrix, can't work well under this situation. SocialMF model considers trust relationships to the user model. NHPMF model considers similarity relationships to the user model. However, our STPMF model simultaneously consider complementary roles of similarity relationships and trust relationships to the user model by means of a weight w, and captures more information from the data. Thus, if a user with very few ratings, we can recommend items to him/her by means of similarity relationships and trust relationships.

In Fig. 9, we observe that our STPMF method consistently performs better than other methods, especially when few user ratings are given. We believe this is mainly due to that we simultaneously consider complementary roles of similarity relationships and trust relationships to the user model by means of a weight w, and propagate similarity relationships and trust relationships in our model.

G. The Analysis of Propagation

We choose Last.fm dataset for this experiment. The parameter settings of our approach are $\alpha = 0.02$, $w = 0.5$, $\lambda_1 = 0.05$, $\lambda_2 = 0.01$. If we propagate the STPMF algorithm, then we call STPMF1, otherwise STPMF2. Similarly, SocialMF and STPMF and so on. The experimental results as follows:

TABLE VII. PERFORMANCE COMPARISONS FOR ALGORITHM PROPAGATION OR NOT

Model	**K=5,D=10**		**K=5,D=15**	
	MAE	***RMSE***	***MAE***	***RMSE***
UKNN	0.8926	0.9124	0.8926	0.9124
IKNN	0.8913	0.9218	0.8913	0.9218
PMF	0.6765	0.7392	0.6392	0.7101
SocialMF1	**0.6582**	**0.7255**	**0.6161**	**0.6923**
SocialMF2	0.6672	0.7321	0.6200	0.6963
NHPMF1	**0.6508**	**0.7196**	**0.6199**	**0.6963**
NHPMF2	0.6580	0.7251	0.6238	0.7037
STPMF1	**0.6142**	**0.6918**	**0.5870**	**0.6693**
STPMF2	0.6271	0.7164	0.5921	0.6792

From the comparison, we have the following observations:

- For $K = 5, D = 10$, compared with no propagation, SocialMF, NHPMF and STPMF improve the MAE of the results by about 0.9%, 0.72%, 1.29%, and the RMSE of the results by about 0.66%, 0.55%, 2.46%.
- For $K = 5, D = 15$, compared with no propagation, SocialMF, NHPMF and STPMF improve the MAE of the results by about 0.39%, 0.39%, 0.51%, and the RMSE of the results by about 0.4%, 0.74%, 0.99%.

Based on the above analysis, we can conclude that similarity propagation and trust propagation to a certain extent can further improve the quality of recommendation.

VII. CONCLUSION

In this paper, our STPMF model effectively combines the advantages of neighborhood model and matrix factorization model, where complementary roles of similarity relationships and trust relationships to the user model by means of a weight w are considered simultaneously. Furthermore, we propagate similarity relationships and trust relationships to alleviate the data sparsity and cold-start problems. Experimental results on two real world datasets show that our model outperforms state-of-the-art collaborative filtering methods, and alleviates the problems of collaborative filtering. With the rapid development of social networks, we can further make full use of social data to enhance recommendation accuracy, such as time information [26][27], content information [28][29] and so on. In the future, we plan to fuse time or content information into probabilistic matrix factorization model, and build user or item models through deep learning. Thus, we can capture more latent information from the data so as to make better recommendations.

ACKNOWLEDGMENT

We thank the anonymous reviewers for the detailed comments. We thank Di Guo and Jingde Hou for helpful discussions related to the paper. This work is partially supported by China National Science Foundation (Granted Number 61272438, 61202376, 61472253, 61572325). Science and Technology Commission of Shanghai Municipality (14511107702). Shanghai Municipal Education Commission (13ZZ112, 13YZ075, GCZX14014, C14001).

REFERENCES

[1] G. Linden, B. Smith, and J. York. Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Computing, vol. 7, no. 1, pp. 76–80, 2003.

[2] J. Bennet and S. Lanning. The Netflix Prize, KDD Cup and Workshop, 2007. www.netflixprize.com.

[3] K. Ali and W. van Stam. TiVo: Making Show Recommendations Using a Distributed Collaborative Filtering Architecture. Proc. 10th ACM SIGKDD Int. Conference on Knowledge Discovery and Data Mining, pp. 394–401, 2004.

[4] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In Proceedings of ACM SIGIR, pages 230–237, 1999.

[5] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. InProceedings of ACM WWW, pages 285–295, 2001.

[6] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. IEEE Computer, pages 30–37, 2009.

[7] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. Advances in neural information processing systems, pages 1257–1264, 2008.

[8] Yehuda Koren. Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model. ACM SIGKDD international conference on Knowledge discovery and data mining, 2008, 426-434.

[9] R. M. Bell and Y. Koren. Lessons from the netflix prize challenge. ACM SIGKDD Explorations Newsletter, pages 75–79, 2007.

[10] P. Heymann, G. Koutrika, and H. Garcia-Molina. Can social bookmarking improve web search?. in Proceedings of the International Conference on Web Search and Data Mining, 2008, pp. 195–206.

[11] X. Li, L. Guo, and Y. E. Zhao. Tag-based social interest discovery. In Proceedings of the 17th International Conference on World Wide Web, 2008, pp. 675–684.

[12] K. H. L. Tso-Sutter, L. Balby Marinho, and L. SchmidtThieme. Tag-aware recommender systems by fusion of collaborative filtering algorithms, in SAC, R. L. Wainwright and H. Haddad, Eds. ACM, 2008.

[13] J. Gemmell, T. Schimoler, B. Mobasher, R. Burke. Resource recommendation for social tagging: a multi-channel hybrid approach. In Proceedings of the 2010 ACM Conference on Recommender Systems, 2010, pp. 60–67.

[14] Wu, L., Chen, E., Liu, Q., Xu, L., Bao, T., & Zhang, L. (2012, October). Leveraging tagging for neighborhood-aware probabilistic matrix factorization. In Proceedings of the 21st ACM international conference on Information and knowledge management (pp. 1854-1858). ACM.

[15] Z. Wang, Y. Wang, and H. Wu. Tags meet ratings：Improving collaborative filtering with tag-based neighborhood method. In IUI’10: Workshop on Social Recommender Systems, 2010.

[16] H. Ma, T. CH. Zhou, M.R. Lyu, I. King. Improving recommender systems by incorporating social contextual information, ACM Transactions on Information Systems 29 (2) (2011). Article 9.

[17] S. Tan, J. Bu, CH. Chen, X. He. Using rich social media information for music recommendation via hypergraph model, ACM Transactions on Multimedia Computing, Communications, and Applications 7 (1) (2011). Article 7.

[18] Mohsen Jamali, Martin Ester. A Matrix Factorization Technique with Trust Propagation for Recommendation in Social Networks. RecSys2010, September 26–30, 2010.

[19] D.H. Lee, P. Brusilovsky. Does trust influence information similarity?. In Proceedings of the 2009 ACM Conference on Recommender Systems, 2009, pp. 71–74.

[20] P. Massa and P. Avesani. Trust-aware recommender systems. In RecSys 2007, USA.

[21] G. Takacs, I. Pilaszy, B. Nemeth and D. Tikk. Matrix factorization and neighbor based algorithms for the Netflix Prize problem. Proc. 2008 ACM Conference on Recommender Systems (RECSYS’08), pp. 267–274, 2008.

[22] T. C. Zhou, H. Ma, I. King, and M. R. Lyu. TagRec: Leveraging tagging wisdom for recommendation. InProceedings of IEEE CSE, pages 194–199, 2009.

[23] H. MA. Learning to recommend. Ph.D. dissertation. The Chinese University of Hong Kong, December 2009.

[24] H. Ma, H. Yang, M. R. Lyu, and I. King. SoRec: Social recommendation using probabilistic matrix factorization. In Proc. of CIKM ’08, pages 931–940, New York, NY, USA, 2008. ACM.

[25] H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In SIGIR 2009, pages 203–210.

[26] Yehuda Koren. Collaborative Filtering with Temporal Dynamics. ACM 2009 Article, 2009.

[27] Liang Xiang, Qing Yang. Time-dependent models in collaborative filtering based recommender system. In: Web Intelligence, pp. 450–457 (2009).

[28] Aaron van den Oord, Sander Dieleman, Benjamin Schrauwen. Deep content-based music recommendation. Advances in Neural Information Processing Systems 26 (NIPS 2013).

[29] Xinxi Wang, Ye Wang. Improving Content-based and Hybrid Music Recommendation using Deep Learning. Proceedings of the ACM International Conference, 2014.