FTMF: Recommendation in Social Network with Feature Transfer and Probabilistic Matrix Factorization

Zhi-Lin Zhao*, Chang-Dong Wang*‡, Yuan-Yu Wan*, Jian-Huang Lai* and Dong Huang†
*School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China.
Guangdong Key Laboratory of Information Security Technology, Guangzhou, China.
Email: zhaozhl7@mail2.sysu.edu.cn, changdongwang@hotmail.com, wanyy6@mail2.sysu.edu.cn, stsljh@mail.sysu.edu.cn
†College of Mathematics and Informatics, South China Agricultural University, Guangzhou, China.
Email: huangdonghere@gmail.com

***Abstract*—It is well-known that recommendation system which is widely used in many e-commerce platforms to recommend items to the right users suffers from data sparsity, imbalanced rating and cold start problems. Matrix factorization is a good way to deal with the sparsity and imbalance problems, which is however unable to make prediction for new users due to the lack of auxiliary information. With the advent of online social networks, the trust relation in the social network can be utilized as auxiliary data to solve the aforementioned problems since consumers' buying behavior is usually affected by the people around them. This paper reports a study of exploiting the trust relationship in social network for personalized recommendation. Although previous studies have paid attention to this topic, we improve the quality of recommendation further and solve the cold start problem better. To this end, we propose a recommendation system in social network with Feature Transfer and Probabilistic Matrix Factorization (FTMF). The auxiliary data and matrix factorization technique are integrated to learn a social latent feature vector of users which represents the features transferred from trusted people. And an adaptive firm factor is introduced to balance the impact from user's own factors and trusted people on buying behavior for each user. The experimental results show that our model can effectively use the auxiliary data and outperforms the existing state-of-the-art social network based recommendation algorithms.**

I. Introduction

In the era of information explosion, it is difficult for us to select useful information from large volumes of data before we make a decision in a short time. So the recommendation system came into being. Nowadays, recommendation system has been one of the most useful tools which help users get interesting information in a short time. More and more e-commerce and video sharing websites like Amazon [1], Twitter [2] and YouTube [3] develop recommendation system to promote sales and improve user experience. Recommendation system usually makes prediction and recommendation by analyzing historical records of users' purchasing and rating. Collaborative filtering [4], [5] is one of the most successful technologies in personalized recommendation. Assuming that similar users would prefer similar items, the algorithm using the large amount of rating records from users to predict which items the target user will like has been extensively studied recently [6], [7], [8]. But the algorithm suffers from some serious problems, namely the sparsity problem, imbalance of rating data, and cold start problem i.e. it is difficult for us to recommend items to the new user or the user rarely left ratings.

Matrix factorization technique [9] is one of the most widely used methods for solving the data sparsity and imbalance problems. Matrix factorization technique usually learns the latent features of both users and items from the sparse user-item rating matrix, and then predicts the ratings to non-purchased items according to user and item latent features. Although this kind of methods have achieved remarkable results, they still suffer from the cold start problem due to lack of auxiliary information for the new users or the users rarely left ratings.

Many social networking sites such as Facebook, and Twitter [10] have become a good platform for us to meet people and communicate with each other. And we can extract the direct trust network [11] from all the users' friend lists in the social network which is an available auxiliary data to solve the cold start problem and improve the quality of recommendation. Because most of users' buying behaviors will be affected by the people around and users usually consider both their own needs and the opinions from trusted people to make shopping options. Traditionally, we just use the similarities calculated by the rating records to measure the relationship between users. Due to social network, we can fuse the trust information and rating records to measure the relationship between users preferably, and then improve the quality of recommendation.

Although some previous studies have paid attention to auxiliary data, they usually directly use the trust information and think that if one user trusts his friends, he would like the items his trusted people like [12], [13]. In fact, users may not like the items though the trusted people like them and the trusted people may not affect the target user at all. We focus on how to mine the value of precious auxiliary data adequately. We propose a recommendation system in

‡Corresponding author

978-1-5090-0620-5/16/$31.00

social network with Feature Transfer and Probabilistic Matrix Factorization (FTMF) which adaptively considers the influence from the trusted people as well as the firm factor of the target user and makes a predicted rating to a non-purchased item to the user. The contributions are summarized as follows:

1) We fuse the trust information in social network and rating records by designing a new algorithm based on probabilistic matrix factorization to learn item latent feature vector, user personal latent feature vector, user social latent feature vector which represents the feature transferred from user's trusted people.
2) Feature Transfer is used to get the social influence from trusted people. For making an adaptive distinction between impressionable and firm people, we will learn an adaptive firm factor which varies from one user to another to trade-off the personal and social influences in making decisions. We think that firm people may not be affected by trusted people easily and vice versa.
3) We conduct experiments on several real world datasets associated with social network and the results show that our model can effectively use the auxiliary data and outperforms the existing social network based recommendation systems.

II. The Proposed Algorithm

In this section, we propose a recommendation algorithm which incorporates user trust relationship in social network based on probabilistic matrix factorization.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
u_1	5	0	3	2	0	0	3
u_2	0	4	0	1	0	0	2
u_3	4	0	3	0	5	2	0
u_4	0	5	4	1	0	0	0
u_5	0	0	0	5	0	0	0
u_6	1	0	0	0	4	4	0

Fig. 1: User-item rating matrix.

A. Probabilistic Matrix Factorization

In recommendation algorithm, a user-item rating matrix $R = [r_{ij}]_{m \times n}$ as shown in Fig. 1 will be used to represent the rating relation between m users and n items. Each entry r_{ij} in line i and column j denotes the rating of user i to item j within a certain numerical interval $[R_{min}, R_{max}]$ which will vary in different datasets. In this paper, without loss of generality, we map all the ratings to the interval $[0, 1]$ using the function $f(x) = (x - R_{min})/(R_{max} - R_{min})$. The higher the rating value r_{ij} is, the more user i is satisfied with item j and if user i has not yet rated item j, the value r_{ij} will be set to zero. The matrix factorization can be used to factorize the user-item rating matrix $R \in \mathbb{R}^{m\times n}$ into two matrixes $U \in \mathbb{R}^{m\times l}$ and $V \in \mathbb{R}^{n\times l}$, with l-dimensional row vectors U_i and V_j representing user latent feature vector of user i and latent feature vector of item j respectively. The conditional probability of the observed ratings is defined as:

$$p(R|U,V,\sigma_R^2) = \prod_{i=1}^{m}\prod_{j=1}^{n}\Big[\mathcal{N}\big(r_{ij}|g(U_iV_j^T),\sigma_R^2\big)\Big]^{I_{ij}^R} \tag{1}$$

where $\mathcal{N}(x|\mu,\sigma_R^2)$ is the normal distribution with mean μ and variance σ_R^2, and I_{ij}^R is the indicator function that is equal to 1 if user i has rated item j or 0 otherwise. The function $g(x)$ is the logistic function $g(x) = 1/(1+e^{-x})$, which can bound the range of $U_iV_j^T$ within [0,1]. Zero means Gaussian priors are assumed for user and item feature vectors:

$$\begin{aligned} p(U|\sigma_U^2) &= \prod_{i=1}^{m}\mathcal{N}(U_i|0,\sigma_U^2\mathbf{I}) \\ p(V|\sigma_V^2) &= \prod_{j=1}^{n}\mathcal{N}(V_j|0,\sigma_V^2\mathbf{I}) \end{aligned} \tag{2}$$

According to the Bayesian inference, the posterior probability of the latent variables U and V can be obtained as follows:

$$\begin{aligned} p(U,V|R,\sigma_R^2,\sigma_U^2,\sigma_V^2) &\propto p(R|U,V,\sigma_R^2)p(U|\sigma_U^2)p(V|\sigma_V^2) \\ &= \prod_{i=1}^{m}\prod_{j=1}^{n}\Big[\mathcal{N}\big(r_{ij}|g(U_iV_j^T),\sigma_R^2\big)\Big]^{I_{ij}^R} \\ &\times \prod_{i=1}^{m}\mathcal{N}(U_i|0,\sigma_U^2\mathbf{I}) \times \prod_{j=1}^{n}\mathcal{N}(V_i|0,\sigma_V^2\mathbf{I}) \end{aligned} \tag{3}$$

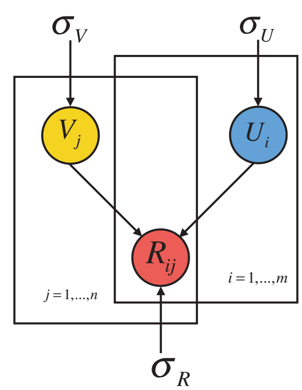

Fig. 2: The graphical model for Probabilistic Matrix Factorization .

The corresponding graphical model is shown in Fig. 2, which is the model of Probabilistic Matrix Factorization (PMF). According to Eq. (3), the latent feature vectors of users and items can be learned from the user-item rating matrix.

B. User Social Latent Feature

In online social networks, user relationship can be represented by a direct trust network $\mathcal{G}(\mathcal{U},\mathcal{E})$ as shown in Fig. 3, where the vertex set $\mathcal{U} = \{u_i\}_{i=1}^m$ represents all the users in the social network and edge set $\mathcal{E}$ denotes the trust relation from a user to another user. In this paper, a binary adjacent matrix $S = [s_{ik}]_{m\times m}$ which is also called trust matrix will be utilized to represent the graph $\mathcal{G}$. It is worth noting that matrix S is an asymmetric matrix since in the real social network, we can't deduce that user k trusts user i even if user i has already trusted user k. For each pair of users, u_i and u_k, the value of s_{ik} is equal to 1 if user u_i trusts user u_k or 0 otherwise.

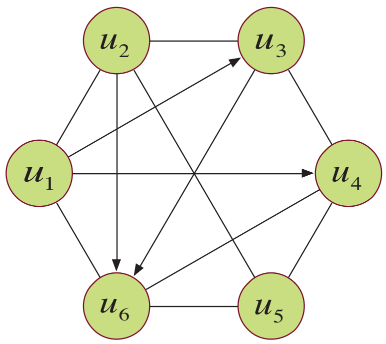

Fig. 3: Trust network.

It is obvious that the users trusted by the target user can make an impact on his buying behavior. People may be willing to buy the item that his trusted users have bought or ask his trusted users for advice before making choices. For each user i, we express his *personal* latent feature by a vector U_i and the *social* latent feature learned from his trusted people by a l-dimensional vector Z_i which is a row vector of the matrix $Z \in \mathbb{R}^{m\times l}$.

Due to the feature transfer, the behavior of user i can be affected by his direct trusted users. So the social latent feature of user i relies on the personal latent feature of all his direct trusted users which means the personal latent features from trusted people of user i can be transferred to that of the user. In other words, the value of social latent feature matrix Z can be derived by personal latent feature matrix U of trusted people and trust matrix S. This is the underlying rationale for the name "feature transfer" in our algorithm. For each social latent feature vector, the zero means Gaussian prior as shown below can help prevent over-fitting:

$$p(Z|\sigma_Z^2) = \prod_{i=1}^{m} \mathcal{N}(Z_i|0, \sigma_Z^2\mathbf{I}) \tag{4}$$

We can get the average personal latent feature vector $\widehat{U}_i$ of the direct trusted people of the target user i in the social network according to U and S. It can be formulated as follows:

$$\widehat{U}_i = \frac{\sum_{k=1}^m s_{ik}U_k}{\sum_{k=1}^m s_{ik}} \tag{5}$$

Because all nonzero values in the trust matrix S are 1 and it is inconvenient for us to measure the total feature transferred from trusted people, so we normalize each row i of the trust matrix so that $\sum_{k=1}^m S_{ik} = 1, \forall i$. Now, we can rewrite the formula as:

$$\widehat{U}_i = \sum_{k=1}^{m} s_{ik}U_k \tag{6}$$

Notice that, although the social latent feature vector Z_i is based on the average personal latent feature vector $\widehat{U}_i$, it is not necessary for Z_i to be equal to $\widehat{U}_i$. As we know, we can be affected by trusted people but can't become the same as them so the features of the trusted people will not transfer to the target user totally and there must exist something different between Z_i and $\widehat{U}_i$. Now, we formulate the conditional distribution of Z given by personal latent feature (i.e. U) and trust relationship (i.e. S) as follows:

$$\begin{aligned} &p(Z|U, S, \sigma_Z^2, \sigma_S^2) \\ &\quad \propto p(U, S|Z, \sigma_S^2) \times p(Z|\sigma_Z^2) \\ &\quad = \prod_{i=1}^{m}\prod_{j=1}^{n} \mathcal{N}(\sum_{k=1}^{m} s_{ik}U_k|Z_i, \sigma_S^2\mathbf{I}) \times \prod_{i=1}^{m} \mathcal{N}(Z_i|0, \sigma_Z^2\mathbf{I}) \end{aligned} \tag{7}$$

If we consider only the features transferred from trusted people of user i, we can still get a predicted rating to item j by $g(Z_iV_j^T)$ because Z_i obtained from the personal latent feature of trusted users is still a user latent feature essentially. According to the Bayesian inference, the posterior probability of the latent variables Z and V can be obtained as follows:

$$\begin{aligned} p(Z, V|R, \sigma_R^2, \sigma_Z^2, \sigma_V^2) &\propto p(R|Z, V, \sigma_R^2)p(Z|\sigma_Z^2)p(V|\sigma_V^2) \\ &= \prod_{i=1}^{m}\prod_{j=1}^{n}\Big[\mathcal{N}\big(r_{ij}|g(Z_iV_j^T), \sigma_R^2\big)\Big]^{I_{ij}^R} \\ &\times \prod_{i=1}^{m} \mathcal{N}(Z_i|0, \sigma_Z^2\mathbf{I}) \times \prod_{j=1}^{n} \mathcal{N}(V_i|0, \sigma_V^2\mathbf{I}) \end{aligned} \tag{8}$$

which is also the model of Probabilistic Matrix Factorization (PMF) essentially.

C. FTMF

Now we can get two different rating predictions for a target user to an item according to his personal latent features and social latent features respectively. One is the predicted rating by only considering user's personal features and not taking the user's features transferred from the trusted people into account. In real life, some people are determined who are stick to themselves when selecting an item and it is not easy for the opinions from trusted people to shake his decision. However, others do not stand firm and are so easily swayed when trusted people give some advices to him. This leads to another predicted rating by only considering user's features transferred from trusted people. In a word, the same personal and social latent features may bring different impacts on different users.

To adaptively trade-off the personal and social latent features of each user, we design an adaptive firm factor which varies from one user to another. Let $\alpha \in \mathbb{R}^{m\times 1}$ be the firm factor vector with α_i being the strength of firmness of user i while $1 - \alpha_i$ being the effect of all the trusted people of user

i. As we know, no one will just be influenced by others and haven't their own idea at all. Besides, most of users will not ignore all the advices from trusted people. Hence, a Gaussian priors are assumed for each firm factor with $1/2$ mean rather than zero as follows:

$$p(\alpha|\sigma_\alpha^2) = \prod_{i=1}^{m} \mathcal{N}(\alpha_i|1/2, \sigma_\alpha^2) \tag{9}$$

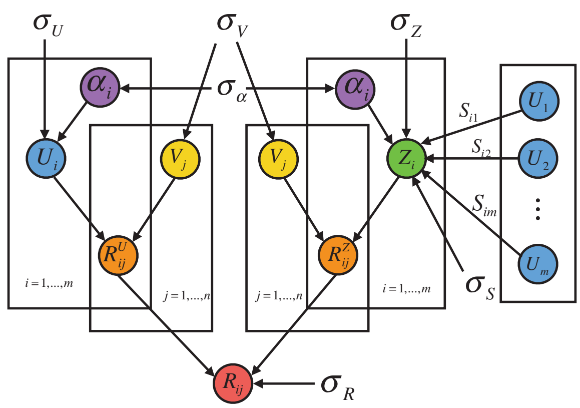

Fig. 4: The FTMF model.

In order to synthesize user personal latent features U, user social latent features Z, item latent features V and firm factors α to make a better recommendation in the social network, we solve the problem of recommendation in social network using the graphical model shown in Fig. 4, which fuses both user-item rating matrix and trust matrix into one objective function. Through a Bayesian inference, we have the following equation for the posterior probability distribution of latent feature vectors and firm factor given by the user-item rating matrix and trust matrix:

$$\begin{aligned}
&p(U, Z, V, \alpha|R, S, \sigma_R^2, \sigma_S^2, \sigma_U^2, \sigma_Z^2, \sigma_V^2, \sigma_\alpha^2) \\
&\quad \propto p(R|U, Z, V, \alpha, \sigma_R^2) \times p(Z|U, S, \sigma_Z^2, \sigma_S^2) \\
&\quad \times p(U|\sigma_U^2) \times p(Z|\sigma_Z^2) \times p(V|\sigma_V^2) \times p(\alpha|\sigma_\alpha^2) \\
&\quad = \prod_{i=1}^{m}\prod_{j=1}^{n}\left[\mathcal{N}\left(r_{ij}|g\left(\left(\alpha_i U_i + (1-\alpha_i)Z_i\right)V_j^T\right), \sigma_R^2\right)\right]^{I_{ij}^R} \\
&\quad \times \prod_{i=1}^{m}\mathcal{N}(\sum_{k=1}^{m} s_{ik}U_k|Z_i, \sigma_S^2\mathbf{I}) \times \prod_{i=1}^{m}\mathcal{N}(U_i|0, \sigma_U^2\mathbf{I}) \\
&\quad \times \prod_{i=1}^{m}\mathcal{N}(Z_i|0, \sigma_Z^2\mathbf{I}) \times \prod_{j=1}^{n}\mathcal{N}(V_i|0, \sigma_V^2\mathbf{I}) \\
&\quad \times \prod_{i=1}^{m}\mathcal{N}(\alpha_i|1/2, \sigma_\alpha^2)
\end{aligned} \tag{10}$$

Our goal is to maximize the log-posterior over latent features and firm factors to estimate parameters. However, maximizing the log-posterior is equivalent to minimizing the following objective function, which is a sum of squared errors with quadratic regularization terms:

$$\begin{aligned}
&\mathcal{L}(R, S, U, Z, V, \alpha) \\
&\quad = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^R\left(r_{ij} - g\left(\left(\alpha_i U_i + (1-\alpha_i)Z_i\right)V_j^T\right)\right)^2 \\
&\quad + \frac{\lambda_S}{2}\sum_{i=1}^{m}\left(\left(\sum_{k=1}^{m} s_{ik}U_k - Z_i\right)\left(\sum_{k=1}^{m} s_{ik}U_k - Z_i\right)^T\right) \\
&\quad + \frac{\lambda_U}{2}\sum_{i=1}^{m} U_i U_i^T + \frac{\lambda_Z}{2}\sum_{i=1}^{m} Z_i Z_i^T \\
&\quad + \frac{\lambda_V}{2}\sum_{j=1}^{n} V_j V_j^T + \frac{\lambda_\alpha}{2}\sum_{i=1}^{m}\alpha_i^2
\end{aligned} \tag{11}$$

where $\lambda_U = \sigma_R^2/\sigma_U^2$, $\lambda_Z = \sigma_R^2/\sigma_Z^2$, $\lambda_V = \sigma_R^2/\sigma_V^2$ and $\lambda_\alpha = \sigma_R^2/\sigma_\alpha^2$. And the gradients of parameters to be estimated have the following forms:

$$\begin{aligned}
\frac{\partial\mathcal{L}}{\partial U_i} &= -\sum_{j=1}^{n}\alpha_i I_{ij}^R\left(r_{ij} - g\left(\left(\alpha_i U_i + (1-\alpha_i)Z_i\right)V_j^T\right)\right) \\
&\quad \times g'\left(\left(\alpha_i U_i + (1-\alpha_i)Z_i\right)V_j^T\right)V_j \\
&\quad + \lambda_S\sum_{k=1}^{m} s_{ki}\left(\sum_{w=1}^{m} s_{kw}U_w - Z_k\right) + \lambda_U U_i.
\end{aligned} \tag{12}$$

$$\begin{aligned}
\frac{\partial\mathcal{L}}{\partial Z_i} &= -\sum_{j=1}^{n}(1-\alpha_i) I_{ij}^R\left(r_{ij} - g\left(\left(\alpha_i U_i + (1-\alpha_i)Z_i\right)V_j^T\right)\right) \\
&\quad \times g'\left(\left(\alpha_i U_i + (1-\alpha_i)Z_i\right)V_j^T\right)V_j \\
&\quad - \lambda_S\left(\sum_{k=1}^{m} s_{ik}U_k - Z_i\right) + \lambda_Z Z_i.
\end{aligned} \tag{13}$$

$$\begin{aligned}
\frac{\partial\mathcal{L}}{\partial V_j} &= -\sum_{i=1}^{m} I_{ij}^R\left(r_{ij} - g\left(\left(\alpha_i U_i + (1-\alpha_i)Z_i\right)V_j^T\right)\right) \\
&\quad \times g'\left(\left(\alpha_i U_i + (1-\alpha_i)Z_i\right)V_j^T\right)\left(\alpha_i U_i + (1-\alpha_i)Z_i\right) \\
&\quad + \lambda_V V_j.
\end{aligned} \tag{14}$$

$$\begin{aligned}
\frac{\partial\mathcal{L}}{\partial \alpha_i} &= -\sum_{j=1}^{m} I_{ij}^R\left(r_{ij} - g\left(\left(\alpha_i U_i + (1-\alpha_i)Z_i\right)V_j^T\right)\right) \\
&\quad \times g'\left(\left(\alpha_i U_i + (1-\alpha_i)Z_i\right)V_j^T\right)\left((U_i - Z_i)V_j^T\right) \\
&\quad + \lambda_\alpha \alpha_i
\end{aligned} \tag{15}$$

where $g'(x) = e^x/(1+e^x)^2$ is the derivative of logistic function.

A local minimum of the objective function can be found by performing gradient descent in latent feature vectors U_i,

Z_i, V_j and firm factors α_i iteratively, as summarized in Algorithm 1. The learning rate μ is a tuneable parameter

Algorithm 1 Parameter Estimation

```
Input: user-item rating matrix: R, trust network: S
select a learning rate μ .
select regularization coefficient λ_S, λ_U, λ_Z, λ_V, λ_α.
select maximum number of iteration MaxIter.
set the starting values of matrixes U, Z, V, α.
for i = 1 to MaxIter do
    Compute gradients ∇_U, ∇_Z, ∇_V and ∇_α
    Set U = U − μ∇_U
    Set Z = Z − μ∇_Z
    Set V = V − μ∇_V
    Set α = α − μ∇_α
end for
Output: U, Z, V, α
```

which will affect the learning convergence. If the learning rate is too large, the solutions of parameters will be divergent. On the contrary, it will take long time until the solution is convergent if the learning rate is too small. So we should use an appropriate learning rate through prior knowledge to get a good performance of the solutions.

The function predicting the rating p_{ij} of a target user i to a non-purchased item j is:

$$p_{ij} = g\Big(\big(\alpha_i U_i + (1-\alpha_i) Z_i\big) V_j^T\Big) \tag{16}$$

where p_{ij} is an element of the predicted user-item rating matrix $P \in \mathbb{R}^{m \times n}$. And we must ensure that all predicted ratings p_{ij} are in an interval $[f(R_{min}), f(R_{max})]$. When the predicted rating goes beyond the range, some adjustments must be applied:

$$p_{ij} = \begin{cases} f(R_{max}), & \text{if } p_{ij} > f(R_{max}) \\ f(R_{min}), & \text{if } p_{ij} < f(R_{min}) \\ p_{ij}, & \text{otherwise} \end{cases} \tag{17}$$

III. Experiment

In this section, extensive experiments have been conducted on the Ciao, Epinions and Flixster datasets [14] to evaluate the effectiveness of our social recommendation algorithm. First, we compare the proposed FTMF algorithm with 10 existing algorithms, the comparison results of which show that our method significantly outperforms the existing algorithms in the challenging recommendation tasks. Then, we also conduct parameter analysis to show the robustness of our method.

A. Description of Datasets

Our proposed algorithm can utilize both rating records and trust records to make a better recommendation. Hence, in this paper, we evaluate our method on three widely tested real-world datasets: Ciao, Epinions and Flixster. All of those datasets contain not only user rating records but also user trust records. The scale of those datasets are shown in Table I. All of them are popular product review sites in which users can write reviews and rate for some items. Besides, users can meet others with similar tastes and try to trust them in those sites. Each rating record in those datasets contains [User_ID, Item_ID, Rating] and all the ratings to items are restricted in a certain numerical interval $[1, 5]$. Each trust record in those datasets contains [User_ID1, User_ID2] which means the former trusted the latter. Note that the trust records are asymmetrical, the former may trust the latter, but the reverse may not be true. Preparing data for experiments, the user-item rating matrix and user direct trust network can be constructed according to the rating records and the trust records respectively.

TABLE I: Datasets.

Statistics	Ciao	Epinions	Flixster
#users	7375	22166	147612
#items	106797	296277	48794
#ratings	284086	922267	8196077
#trusts	111781	355754	7058819

B. Evaluation Metrics

In order to evaluate the quality of the recommendation algorithms, two widely used evaluation metrics, namely Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) [15], will be used to measure the accuracy of the ratings predicted by our algorithm and other state-of-the-art algorithms. The metrics of MAE and RMSE are defined as:

$$MAE = \frac{1}{T} \sum_{i,j} |r_{ij} - \widehat{r}_{ij}| \tag{18}$$

$$RMSE = \sqrt{\frac{1}{T} \sum_{i,j} (r_{ij} - \widehat{r}_{ij})^2} \tag{19}$$

Here r_{ij} denotes the rating user i given to item j, $\widehat{r}_{ij}$ denotes the rating user i given to item j as predicted by a recommendation algorithm and T denotes the number of tested ratings. From the definitions, we know that a smaller MAE or RMSE value means better prediction quality of an algorithm.

C. Comparison Experiments

In this section, in order to show the effectiveness of our proposed social recommendation algorithm, we compare the recommendation results with those generated by the state-of-the-art recommendation algorithms as follows.

1) **UBCF** [16]: The user-based collaborative filtering algorithm tries to find out some similar users to the target user and recommend some items to the target user according to what they have purchased.
2) **IBCF** [17]: The item-based collaborative filtering algorithm will recommend some non-purchased items which are similar to those items the target user has bought.
3) **TidalTrust** [12]: Considering all the users at the shortest distance from the target user, the algorithm makes a rating prediction to a non-purchased item for the target user according to the ratings to the item given by his direct trusted users.

4) **MoleTrust** [13]: Similar to the idea of TidalTrust, the algorithm considers the ratings to non-purchased items from all the trusted users up to a maximum-depth d in the trust network to make a prediction for the target user. We use the best $d = 5$ in our experiments.
5) **PMF** [18]: The algorithm, which only uses user-item rating matrix, extracts user and item latent feature vectors by decomposing the matrix and makes rating prediction according to those latent features.
6) **SoRec** [19]: The algorithm is a factor analysis approach by using both users' trust information and rating records based on probabilistic matrix factorization. Each item is represented by a latent factor but each user is represented by two factors, one for the initiator and the other for the receiver.
7) **SR** [20]: Based on matrix factorization with social regularization, the algorithm tries to improve the quality of recommendation by incorporating social network information and the social regularization term by adopting an assumption that the more similar between a user and a friend of him, the more close taste they should have.
8) **socialMF** [21]: The algorithm, which is a model-based approach for recommendation in social network, supposes that the latent features of a user depend on those of his direct trusted users, and employs matrix factorization techniques to get user and item latent features.
9) **RSTE** [22]: The probabilistic factor analysis framework which considers the connections among users fuses the users' tastes and their trusted friends' favors together in a nature way.
10) **FIP** [23]: Similar to SoRec, the algorithm is also a factor analysis approach which factorizes both rating matrix and the trust network. But the difference is that each user is represented by only one factor. Besides, user and item features are used as priors for user and item factors respectively.

All the above algorithms can be simply divided into two types, one type is non-matrix factorization method such as UBCF, IBCF, TidalTrust and MoleTrust and the other type is matrix factorization method such as PMF, SoRec, SR, socialMF, RSTE, FIP and FTMF. To be fair, we set dimensionally $l = 10$, learning rate $\mu = 0.01$ and the max number of iterations $MaxIter = 100$ for all the matrix factorization methods. The parameter settings of our approach are $\lambda_U = \lambda_Z = \lambda_V = \lambda_\alpha = 0.01$ and $\lambda_S = 0.1$. For each dataset, 90% data are taken as training data and the remaining 10% as testing data.

1) Performance on all users: The comparison results in terms of MAE and RMSE on the three datasets are reported in Fig. 5, Fig. 6 and Fig. 7 respectively. The plots clearly show that all the non-matrix factorization methods namely, UBCF, IBCF, TidalTrust and MoleTrust have the similar capability to make a prediction on all the datasets but the effectiveness of UBCF and IBCF which only utilize rating records are slightly worse than the other two algorithms which utilizes user trust information, TidalTrust and MoleTrust, in general. As we can see, with the increase of iterations, the performance of all the matrix factorization methods will be better and better, and all the final results of those matrix factorization methods are quite good. But the effectiveness of the RSTE algorithm is close to the other non-matrix factorization methods on Ciao and Flixster datasets and is poor on Epinions dataset. Besides, among the three algorithms which just use user-item rating matrix like UBCF, IBCF and PMF, the PMF works significantly better than the other two algorithms. It means that matrix factorization technique is a good way to extract user and item latent features to make a better rating prediction. Therefore, if we integrate user trust information to matrix factorization to get improved variant like SoRec, FIP, socialMF, RSTE, SR and FTMF, the performance can be further improved in most of cases on all the datasets. The comparison results show that the information of user trust network can be used to improve the accuracy of rating prediction. Above all, it is obvious that the proposed FTMF algorithm can get the best results compared with other algorithms on the three datasets in any case. Besides, our algorithm can almost get the best result in each iteration compared with other matrix factorization methods which means that our algorithm has a faster convergence velocity than the other algorithms.

TABLE II: Comparison results for cold start users.

Dataset	Ciao		Epinions		Flixster	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
UBCF	0.772	1.047	0.877	1.086	0.892	1.306
IBCF	0.772	1.047	0.877	1.096	0.942	1.335
TidalTrust	0.773	1.043	0.842	1.092	0.937	1.319
MoleTrust	0.767	1.032	0.823	1.084	0.901	1.314
PMF	0.764	1.021	0.816	1.063	0.925	1.260
SoRec	0.770	1.035	0.756	1.026	0.819	1.150
FIP	0.766	1.026	0.743	1.022	0.897	1.225
socialMF	0.768	1.029	0.788	1.034	0.871	1.194
RSTE	0.771	1.042	0.797	1.044	0.922	1.228
SR	0.767	1.023	0.807	1.054	0.901	1.223
FTMF	**0.762**	**1.020**	**0.728**	**0.988**	**0.773**	**1.051**

2) Performance on cold start users: In the social network, some users have rated many items but most of users express a few ratings. We consider users who have not rated any items at all and rated less than 10 items as cold start users. Actually, it is difficult for us to make a rating prediction for those users since we have few relevant information which will lead to an unreliable recommendation. So the performance of recommendation for cold start users is of great importance. In the social network, the trust information can be used as auxiliary data to solve the cold start problem. The results in terms of MAE and RMSE on the three datasets for cold start users only are shown in TABLE II. As we can see, the algorithm using the information in social network such as TidalTrust and MoleTrust are better than the others in the non-matrix factorization methods in most of cases. In the matrix factorization methods, almost all the algorithms combining the trust network are better than the PMF which just uses user-item rating matrix. Compared with other recommendations, the proposed FTMF algorithm can make the most accurate rating prediction for cold start users which is mainly due to

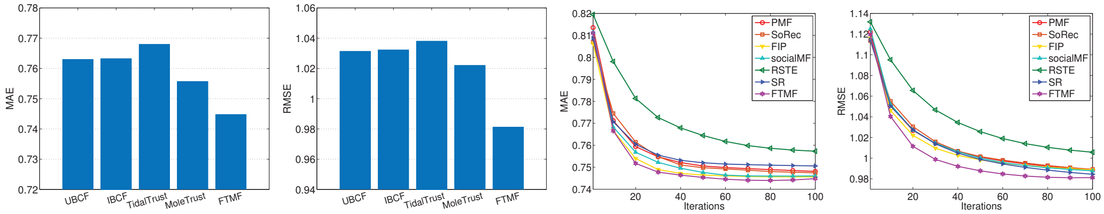

Fig. 5: Comparison results on the Ciao dataset.

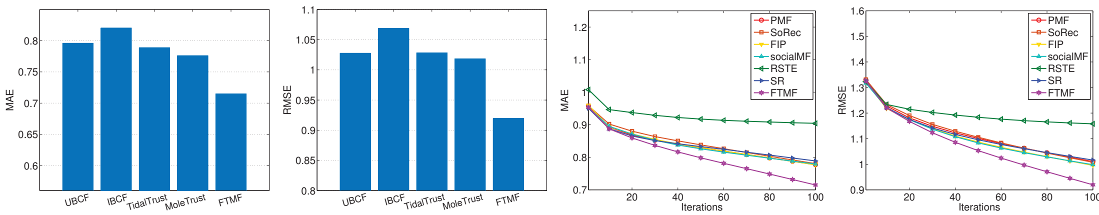

Fig. 6: Comparison results on the Epinions dataset.

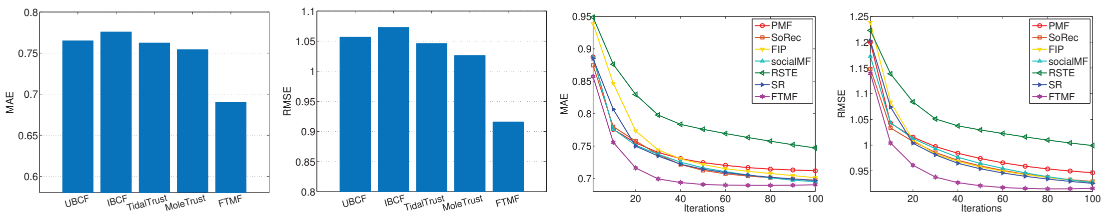

Fig. 7: Comparison results on the Flixster dataset.

not only the feature transferred from trusted people but also the firm factor which adaptively balances the personal features and social features of each user.

D. Parameter Analysis

Parameter λ_S controls the feature transferred from the trusted users to the target user in the social network. Larger value of λ_S indicates the more features will transfer to the target users. And the value of dimensionality l represents the number of latent features of users and items. Theoretically, a larger l value will lead to a more accurate result since we can use more features to represent users and items. In this section, we use different amounts of training data (30%, 60%, 90%) to analyze the impact of different values of parameters to the result in our algorithm.

Fig. 8, Fig. 9 and Fig. 10 show the impacts of λ_S and l in terms of MAE and RMSE on three datasets respectively. We obverse that the performance of the algorithm will be improved as the amounts of training data increases on Ciao and Flixster datasets. However, the results on the Epinions dataset with 30% and 60% as training data are almost the same. Generally speaking, the larger portion the training set is, the better the performance of the algorithm is because we have more data to construct a more accurate model. If we increase the value of l from 5 to 10 and keep other parameters fixed, the predicted results on Ciao dataset will be more accurate but there are almost no influence on the other two datasets which means that it is enough for us to get the best results on the two datasets when the dimensionality value $l = 5$. When we vary the regularization coefficient λ_S from 0.01 to 0.1 with the step size 0.01, the test results are stable and will not vary acutely, especially on the Epinions dataset, which confirms that our algorithm is quite robust to the parameter.

IV. Conclusion and Future Works

Despite the great success of the recommendation algorithm in plenty of real-world applications, recommendation system still suffers from data sparsity, imbalance and cold start problems. To address these problems, in this paper, we have proposed a novel method, which is based on social network with Feature Transfer and probabilistic Matrix Factorization (FTMF). By utilizing probabilistic matrix factorization, we integrate the trust matrix in social network and rating matrix to learn the item latent feature vectors, user personal latent feature vectors and user social latent feature vectors simultaneously. Different from the existing models, our model can make a distinction between impressionable and firm people by designing an adaptive firm factor. Experiments on three real-world datasets namely Ciao, Epinions and Flixster show that our method outperforms the existing methods of both traditional recommendation algorithms and social network based recommendation algorithms. Besides, the proposed algorithm deals better with cold start user problem than the existing methods.

For our future work, we are interested in the cross domain recommendation in social network which can utilize the data in an auxiliary domain to make a recommendation in the target domain in social network.

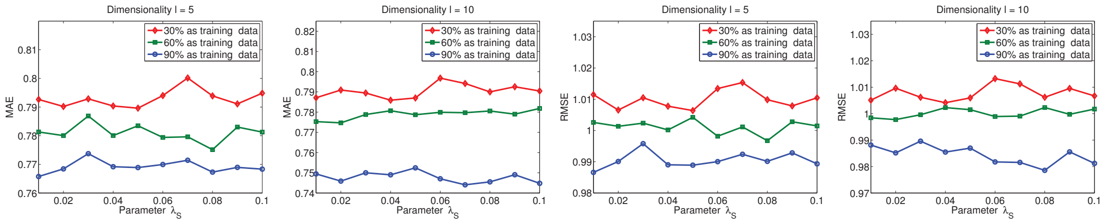

Fig. 8: Parameter analysis of Ciao.

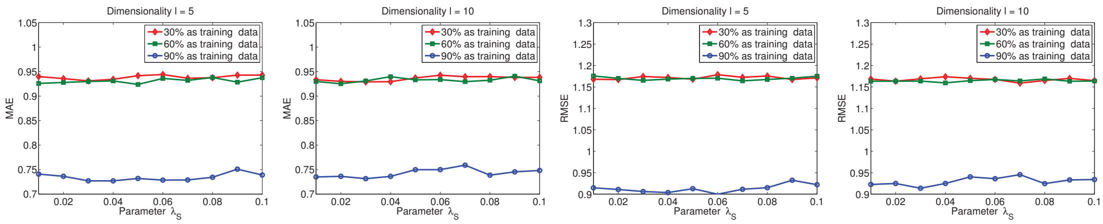

Fig. 9: Parameter analysis of Epinions.

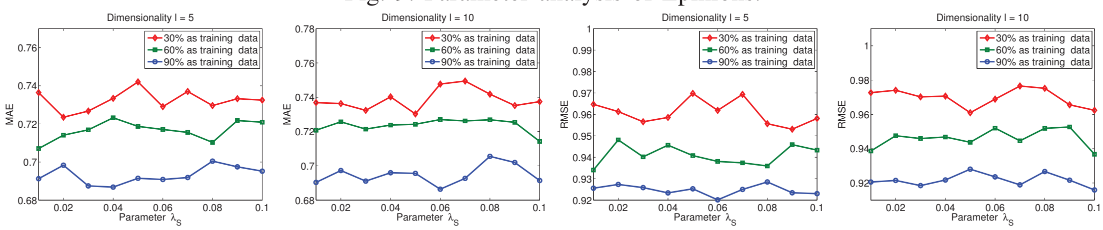

Fig. 10: Parameter analysis of Flixster.

ACKNOWLEDGMENT

This work was supported by NSFC (No. 61502543 and No. 61573387), GuangZhou Program (No. 201508010032), Guangdong Natural Science Funds for Distinguished Young Scholar (16050000051), the PhD Start-up Fund of Natural Science Foundation of Guangdong Province, China (2014A030310180), the Fundamental Research Funds for the Central Universities (16lgzd15).

REFERENCES

[1] G. Linden, B. Smith, and J. York, “Amazon.com recommendations: Item-to-item collaborative filtering,” IEEE Internet Computing, vol. 7, pp. 76–80, 2003.

[2] H. G. Elmongui, R. Mansour, H. Morsy, S. Khater, A. El-Sharkasy, and R. Ibrahim, “TRUPI: Twitter recommendation based on users’ personal interests,” in CICLing, vol. 2, 2015, pp. 272–284.

[3] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. V. Vleet, U. Gargi, S. Gupta, M. L. Yu He, and D. S. Blake Livingston, “The YouTube video recommendation system,” in RecSys, 2010, pp. 293–296.

[4] Y. Shi, M. Larson, and A. Hanjalic, “Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges,” CSUR, vol. 47, pp. 1–45, April 2014.

[5] J. won Lee, H.-J. Kim, and S. goo Lee, “Conceptual collaborative fltering recommendation: A probabilistic learning approach,” Neurocomputing, vol. 73, p. 2793C2796, August 2010.

[6] Y. Chuan, X. Jieping, and D. Xiaoyong, “Recommendation algorithm combining the user-based classified regression and the item-based filtering,” in ICEC, August 2006, pp. 574–578.

[7] J. Zhang, Z. Lin, B. Xiao, and C. Zhang, “An optimized item-based collaborative filtering recommendation algorithm,” in IEEE IC-NIDC, 2009, pp. 414–418.

[8] L. Sa, “Collaborative filtering recommendation algorithm based on cloud model clustering of multi-indicators item evaluation,” in BCGIN, 2011, pp. 645–648.

[9] Y. Koren, R. M. Bell, and C. Volinsky:, “Matrix factorization techniques for recommender systems,” in IEEE Computer, 2009, pp. 30–37.

[10] J. Yang, J. J. McAuley, and J. Leskovec, “Community detection in networks with node attributes,” in ICDM, 2013, pp. 1151–1156.

[11] J. Tang, H. Gao, X. Hu, and H. Liu:, “Exploiting homophily effect for trust prediction,” in WSDMe, 2013, pp. 53–62.

[12] J. A. Golbeck, “Computing and applying trust in web-based social networks,” in PhD thesis, University of Maryland College Park, 2005, pp. 1–199.

[13] P. Massa and P. Avesani, “Trust-aware recommender systems,” in RecSys, 2007, pp. 17–24.

[14] H. Fang, Y. Bao, and J. Zhang, “Leveraging decomposed trust in probabilistic matrix factorization for effective recommendation,” in AAAI, 2014, pp. 30–36.

[15] B. Wang, Q. Liao, and C. Zhang, “Weight based KNN recommender system,” in IHMSC, vol. 2, 2013, pp. 449–452.

[16] J. Wang, A. P. de Vries, and M. J. Reinders, “Unifying user-based and item-based collaborative filtering approaches by similarity fusion,” in SIGIR, August 2006, pp. 501 – 508.

[17] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in WWW, 2001, pp. 285–295.

[18] R. Salakhutdinov and A. Mnih, “Probabilistic Matrix Factorization,” in NIPS, 2007, pp. 1257–1264.

[19] H. Ma, H. Yang, M. R. Lyu, and I. King, “SoRec: Social recommendation using Probabilistic Matrix Factorization,” in CIKM, 2008, pp. 931–940.

[20] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, “Recommender systems with social regularization,” in WSDM, 2011, pp. 287–296.

[21] M. Jamali and M. Ester, “A matrix factorization technique with trust propagation for recommendation in social networks,” in RecSys, 2010, pp. 135–142.

[22] H. Ma, I. King, and M. R. Lyu, “Learning to recommend with social trust ensemble,” in SIGIR, 2009, pp. 203–210.

[23] S. H. Yang, B. Long, and A. Smola, “Like like alike-joint friendship and interest propagation in social networks,” in WWW, 2011, pp. 537–546.