This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2605085, IEEE Transactions on Pattern Analysis and Machine Intelligence

Social Collaborative Filtering by Trust

Bo Yang, Yu Lei, Jiming Liu, and Wenjie Li

Abstract—Recommender systems are used to accurately and actively provide users with potentially interesting information or services. Collaborative filtering is a widely adopted approach to recommendation, but sparse data and cold-start users are often barriers to providing high quality recommendations. To address such issues, we propose a novel method that works to improve the performance of collaborative filtering recommendations by integrating sparse rating data given by users and sparse social trust network among these same users. This is a model-based method that adopts matrix factorization technique that maps users into low-dimensional latent feature spaces in terms of their trust relationship, and aims to more accurately reflect the users reciprocal influence on the formation of their own opinions and to learn better preferential patterns of users for high-quality recommendations. We use four large-scale datasets to show that the proposed method performs much better, especially for cold start users, than state-of-the-art recommendation algorithms for social collaborative filtering based on trust.

Index Terms—Recommender system, collaborative filtering, trust network, matrix factorization.

1 Introduction

MANY previous studies have addressed personalized recommendation systems. Among existing techniques, collaborative filtering (CF) is relatively simple and effective, and has been widely used by many commercial web sites. CF approaches attempt to utilize the available user-item rating data to make predictions about the users preferences. These approaches can be divided into two groups [1]: memory-based (or heuristic-based) and model-based. Memory-based approaches [3], [6], [19], [27], [31], [36] make predictions based on the similarities between users or items, while model-based approaches [9], [12], [20], [32], [39] seek to create a prediction model from rating data through the use of machine learning. In particular, matrix factorization based models [16], [17], [18], [26], [29], [30], [41] have gained popularity in recent years due to their relatively high accuracy and scalability.

Although CF has developed rapidly and achieved huge success over the past two decades, both memory-based and model-based CF approaches have two challenges: data sparsity and cold start, which greatly reduce their performance. Data sparsity occurs when available rating data are very few. When cold start occurs, CF cannot provide satisfactory recommendations to those users who rarely rate items.

One potential solution of these problems is to explore available social networks. With the rapid development of web 2.0 technologies, in addition to the ratings of items contributed by users, the social information of users is much more readily obtained than before through social networking services (see Fig. 1 as an example). It is believed that human beings usually acquire and disseminate information through their acquaintances such as friends, colleagues or partners [10], [15], [28], which implies that the underlying social networks of users might play a fundamental role in helping them filter information. Trust relationship is one of the most important types of social information, because we are more likely to accept viewpoints from those we trust [33], [42]. Thus, it creates a large opportunity, as well as a large challenge, to improve recommendation quality by sufficiently and effectively utilizing available trust information.

- B. Yang is with the School of Computer Science and Technology, Jilin University, Changchun, Jilin, China 130012; Key Laboratory of Symbolic Computation and Knowledge Engineer (Jilin University), Ministry of Education, Changchun, Jilin, China, 130012. E-mail: ybo@jlu.edu.cn.
- Y. Lei and W. Li are with the Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Hong Kong. E-mail: {csylei, cswjli}@comp.polyu.edu.hk.
- J. Liu is with the Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong. E-mail: jiming@comp.hkbu.edu.hk.

1.1 Related Work

Many memory-based approaches [2], [8], [13], [24], [25], [34], [35] have been proposed to directly utilize or explore the available trust networks for use in recommendation systems. Most generate predictions for a target user by aggregating the ratings of his/her direct (sometimes indirect) trusted friends, with the ratings given by users similar to him/her as usual, in different ways. As reported in [13], [25], these methods are able to improve the coverage of recommendations compared against traditional memory-based CF (which do not consider the additional trust information). However, the performance of such methods are largely dependent on the trust propagation models they adopt [34]. In addition, these methods are usually time-consuming, and thus are not suitable for handling large-scale applications because they need to calculate similarities over an entire rating matrix and whole trust network.

Instead of using trust ties to infer users' neighbors and then promote the accuracy of similarity calculation among users, as was done by most memory-based methods, matrix factorization based methods [14], [21], [22], [23], [38], [40] have been proposed to utilize trust information in a different style. These methods simultaneously map users and items into low-dimensional feature spaces, and then train a prediction model by optimizing some objective functions over rating and trust data.

The authors of [21] proposed a probabilistic matrix factorization method, SoRec, that factorizes a social graph

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2605085, IEEE Transactions on Pattern Analysis and Machine Intelligence

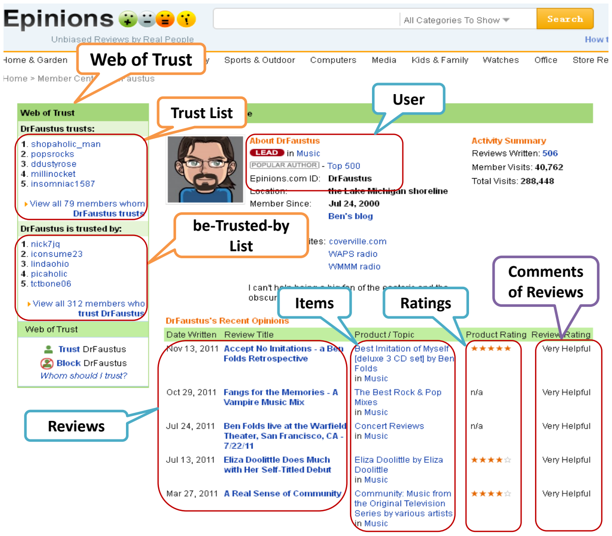


Fig. 1. Epinions: an example of online social networking services.

by $U^{\mathrm{T}}Z$, where U is the user feature matrix, and Z is a factor feature matrix with no realistic implications. They also adopted a function to adjust trust values according to the degrees of involved nodes in the trust network. In [22], the same authors proposed the RSTE method that fuses user i's interests and the interests of his/her trusted friends simultaneously to model a specific rating R_{ij}, i.e., attempts to fit R_{ij} using the following term:

$$(\alpha \cdot U_i + (1-\alpha) \cdot \sum\nolimits_{k \in \mathcal{F}(i)} T_{ik} U_k)^{\mathrm{T}} V_j, \tag{1}$$

where U_i and V_j denote the feature vectors of user i and item j, respectively, $\mathcal{F}(i)$ denotes the set of users who are trusted by user i, T_{ik} is a normalized trust value, and parameter α regulates a trade-off between user i's interest and his/her friends' interest.

A method named TTMF [40] was proposed for tag-based social recommendations by utilizing topic-specific trust relations, where the term "topic" actually refers to a latent factor. For each factor f, TTMF constructs a different trust matrix T^f, and then aggregates all of $T_{ik}^f U_k^f$ as U_i^f (the corresponding dimension in U_i), which is similar to Eq.1.

SocialMF [14] is similar to RSTE, but uses a different implementation to take into account the interests of trusted friends by incorporating a regularization term as follows into its objective function:

$$\beta \cdot \sum_{i=1}^{m} \|U_i - \sum\nolimits_{k \in \mathcal{F}(i)} T_{ik} U_k\|_{\mathrm{F}}^2, \tag{2}$$

where parameter β controls the distance between user i's feature vector and the combinational feature vector of his/her trusted friends.

[23] introduced several social regularization terms that are similar to Eq.2. Each of these works to make user i's feature vector close to his/her trusted friends' feature vectors, and the main difference is that the trust value T_{ik} in social regularization model can be flexibly replaced by a traditional similarity in terms of ratings besides trust/social relationships. The authors of [38] proposed a circle-based method called CircleCon that is an extension of SocialMF. Its basic idea is that the friends of a user may vary with respect to categories of items. According to different categories, it divides an available social network into different partitions, and then adopts the same regularization term in Eq.2 to train different prediction models.

1.2 Motivation and Contribution

For any one user, existing methods utilize the interest of his/her neighbors (i.e., a local structure of trust networks) to model his/her own interest. In doing so, previous matrix factorization based methods take available trust ties as constraints to design objectives, making the interests of a user close to the linear combination of their neighbors' interest during the process of matrix factorization. However, in this work, we utilize the trust ties among users from a new perspective, and attempt to model user's interest more reasonably in terms of his/her own social/trust behaviors, e.g., browsing ratings/reviews, writing ratings/reviews, and adding trust friends, which can be obtained by studying the influence of global trust propagation on the formation of a users' opinions.

Our work is motivated by the observation that users are caught in their social network of mutuality, and their opinions about items will be directly or indirectly affected by others through such a network. More specifically, when a user is rating, he/she will be more likely affected by the existing ratings or reviews provided by others he/she trusts, and in the same way, his/her contributions will consequently have influence on the decisions of others who trust him/her. For instance, in online social networking sites such as Epinions (www.epinions.com, see Fig. 1), when a user P is browsing items (products), if he/she thinks another user Q's opinions (ratings, reviews, or other contents) are relevant and helpful to him/her, he/she would be likely to put Q into his/her own "Trust list". Meanwhile, P's attitude/behavior about those items will be affected by Q according to what extent P trusts Q. Consequently, through this kind of process, P's interest will gradually be affected by others whom he/she trusts, and thus exerts a persistent influence on his/her further decisions. In the same way, P's contributions (ratings, reviews, or other contents) will affect others who appear in P's "be-Trusted-by list" (i.e., who trusts P).

In modeling this mutual influence, we propose a simple but effective way to map users into two low-dimensional spaces, truster space and trustee space, by factorizing trust networks according to the directional property of trust. The vectors of truster (B) and trustee (W) in two spaces describe "to trust by browsing" and "to be trusted by writing" behaviors of a user, respectively. Suppose user P trusts user Q with strength w, then w can be represented as the inner product of P's truster vector (B_P) and Q's trustee vector (W_Q). Viewed from another perspective, the truster vector represents a user's "browsing interest" which essentially catches the influence received by him/her from his/her trusted friends, while the trustee vector represents the user's "writing interest" which essentially catches the influence given by him/her to the friends who trust him/her. Furthermore, the two spaces will be used in tandem with the user space and item space obtained by factorizing rating

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2605085, IEEE Transactions on Pattern Analysis and Machine Intelligence

matrix to construct novel fusing models and to predict the genuine interest of users in a more reasonably way.

Previously, we have published a conference paper to briefly present and preliminarily validate the aforementioned idea [37], which will be extensively extended in this work by supplementing new models, algorithms, theoretical analysis as well as sufficient experimental results. The rest of the paper is organized as follows. In Section 2 we propose a novel matrix factorization based social CF method named TrustMF. In Section 3, we discuss the rationale of TrustMF from a probabilistic view and then contribute a more general framework, named TrustPMF, to predict a user's preferences more effectively by fitting rating data and trust data in an approximate, rather than deterministic, way. In Section 4, we validate the performance of TrustMF and TrustPMF by comparing them with several representative methods on four large real-world datasets.

2 The TrustMF Model

2.1 Factorization of Rating Matrix

Suppose we have a recommender system which involves m users and n items. Let $R = [R_{ij}]_{m \times n}$ denote the observed user-item rating matrix, where each known element R_{ij} denotes the observed rating of item j given by user i, which is normally an integer from 1 to $R_{\max}$ (e.g., 1 to 5), and each unknown element indicates the user has not rated the item yet. By mapping users and items to a low-dimensional feature (factor) space in which each factor can be deemed as any aspect, topic, or other implicit feature relevant to items or users' interests [17], each user or item can be represented by a low-dimensional vector. Let d-dimensional vectors U_i and V_j be the user-specific latent feature vector of user i and item-specific latent feature vector of item j, respectively. Then we have user feature matrix $U \in \mathbb{R}^{d \times m}$ and item feature matrix $V \in \mathbb{R}^{d \times n}$.

By employing low-rank approximation over the observed rating matrix, we can recover unknown ratings by $U^{\mathrm{T}}V$. Formally, the feature matrices U and V can be learned by minimizing a loss function as follows:

$$\mathcal{L} = \sum_{(i,j)\in\Omega} (U_i^{\mathrm{T}} V_j - R_{ij})^2 + \lambda(\|U\|_{\mathrm{F}}^2 + \|V\|_{\mathrm{F}}^2)$$

where $\Omega = \{(i,j) : R_{ij} \neq 0\}$ denotes the locations of observed ratings in rating matrix, $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm, and parameter λ controls the model complexity to avoid over-fitting.

To further avoid over-fitting when learning parameters, a so-called weighted-λ-regularization was introduced to above model [41]. It increases the penalization to the feature vectors of users or items involving more ratings. Correspondingly, the new objective is obtained as follows:

$$\mathcal{L} = \sum_{(i,j)\in\Omega} (U_i^{\mathrm{T}} V_j - R_{ij})^2 + \lambda(\sum_i n_{u_i}\|U_i\|_{\mathrm{F}}^2 + \sum_j n_{v_j}\|V_j\|_{\mathrm{F}}^2)$$

where n_{u_i} and n_{v_j} denote the numbers of ratings given by user i and given to item j, respectively.

2.2 Factorization of Trust Network

Suppose we have a trust network N with m nodes, where nodes represent users and arcs represent directional trust relationships among users. The adjacent matrix $T = [T_{ik}]_{m \times m}$ can be used to describe the structure of N, in which T_{ik} is normally a real number within the interval $[0, 1]$, representing the extent user i trusts k (e.g.,"1" indicates user i extremely trusts k). Since i trusts k does not always means that k trusts i in the same way, T may not be symmetric.

Due to the asymmetric property of trust, in this work we map each user i as two distinct d-dimensional latent feature vectors, depicted by truster-specific feature vector B_i and trustee-specific feature vector W_i, respectively. B_i and W_i characterize the behaviors of "to trust others" and "to be trusted by others" of user i, respectively. Given such vectors, one can model trust value T_{ik} as the inner product of B_i and W_k. It is important to notice that, this way of modeling trust relations is totally distinct from existing methods, which explicitly explores the real-world implications of feature vectors. We will see later this new way of modeling trust values not only characterizes the directional property of trust in a more accurate way, but also gives a better interpretation of how the mutual trust among users are generated and how they affect a user's respective opinions.

Provided we have only trust data, the feature matrices $B \in \mathbb{R}^{d \times m}$ and $W \in \mathbb{R}^{d \times m}$ can be learned by minimizing the objective function:

$$\mathcal{L} = \sum_{(i,k)\in\Psi} (B_i^{\mathrm{T}} W_k - T_{ik})^2 + \lambda(\|B\|_{\mathrm{F}}^2 + \|W\|_{\mathrm{F}}^2)$$

where $\Psi = \{(i,k) : T_{ik} \neq 0\}$ denotes the locations of observed trust relations in the trust matrix.

2.3 TrustMF Model

We have performed matrix factorization just based on either rating data or trust data. In this section, we will present our matrix factorization model to fuse both.

2.3.1 *Truster Model*

As mentioned before, users of social networking such as Epinions are able to browse and generate opinions (ratings or reviews) on items they are interested in and then build their respective trust nets based on such opinions. Through tangled trust ties, the opinions of an individual will be affected by others and vice versa. Here we first propose a truster model to characterize the first aspect, or how others will affect a specific user's opinions.

Note that the m users getting involved in rating matrix R and trust matrix T are the same. Therefore, one can associate R and T into a single matrix factorization process by sharing a common user-specific latent feature space. Fig.2(a) shows the proposed truster model that is capable of characterizing how a user i's opinions in terms of ratings are affected by other users he/she trusts by means of $B_i^{\mathrm{T}} V_j$.

In this model, we choose the truster-specific feature matrix B as the latent space commonly shared by R and T, meaning the user feature matrix U is approximated by truster feature matrix B. Here each vector B_i simultaneously characterizes two aspects: how a user i trusts (or

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2605085, IEEE Transactions on Pattern Analysis and Machine Intelligence

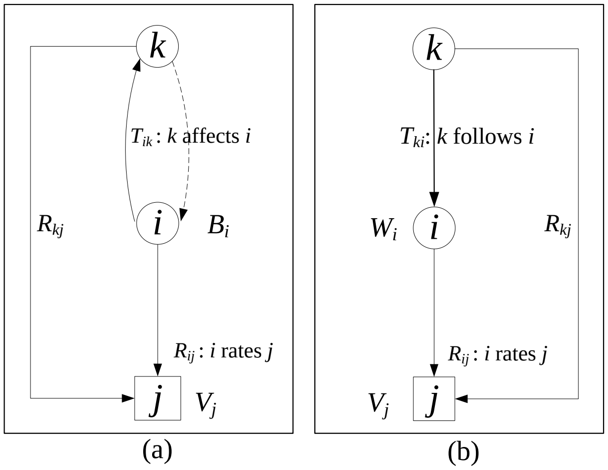


Fig. 2. (a) Truster model $B^T V$: how others affect user i's opinions. (b) Trustee model $W^T V$: how others follow user i's opinions.

is affected by) others and how the same user rates items. Together, the $B_i^{\mathrm{T}} V_j$ indicates how other users affect user i to rate item j, which is the approximation of the real score R_{ij}. Therefore, one can learn the feature matrices B, V and W simultaneously by minimizing the following objective function:

$$\mathcal{L} = \sum_{(i,j)\in\Omega} (B_i^{\mathrm{T}} V_j - R_{ij})^2 + \sum_{(i,k)\in\Psi} (B_i^{\mathrm{T}} W_k - T_{ik})^2 + \lambda(\|B\|_{\mathrm{F}}^2 + \|V\|_{\mathrm{F}}^2 + \|W\|_{\mathrm{F}}^2) \tag{3}$$

In this way, one can integrate two types of data sources and obtain latent spaces B and V that can work together to produce a more accurate prediction.

Since the trust value T_{ik} is between 0 and 1, in order to learn parameters in a more convenient way, we map the raw rating R_{ij} into an interval $(0, 1]$ by employing the function $f(x) = x/R_{\max}$. In order to fit data more conveniently, we adopt the logistic function $g(x) = 1/(1 + \exp(-x))$ suggested by [29] to bound the inner product of latent feature vectors into the interval $[0, 1]$. The prediction can be generated using $g(B_i^{\mathrm{T}} V_j) \cdot R_{\max}$ after model training. In addition, we incorporate a similar regularization approach as weighted-λ-regularization [41] into our model. In summary, the objective function to be optimized in the proposed truster model is as follows:

$$\begin{aligned}\mathcal{L} = &\sum_{(i,j)\in\Omega} (g(B_i^{\mathrm{T}} V_j) - R_{ij})^2 + \sum_{(i,k)\in\Psi} (g(B_i^{\mathrm{T}} W_k) \\ &- T_{ik})^2 + \lambda(\sum_i (n_{b_i} + m_{b_i})\|B_i\|_{\mathrm{F}}^2 \\ &+ \sum_j n_{v_j}\|V_j\|_{\mathrm{F}}^2 + \sum_k m_{w_k}\|W_k\|_{\mathrm{F}}^2)\end{aligned} \tag{4}$$

where n_{b_i} and n_{v_j} denote the number of ratings given by user i and given to item j, respectively, m_{b_i} denotes the number of users who are trusted by user i and m_{w_k} denotes the number of users who trust user k.

2.3.2 Trustee Model

Fig.2(b) shows the proposed trustee model that is able to characterize how a user i's opinions affect the decisions of others who trust i by means of $W_i^T V_j$. Distinct from the truster model, this time we choose the trustee-specific feature matrix W as the latent space commonly shared by R and T, meaning the user feature matrix U is approximated by the trustee feature matrix W. In the trustee model, vector W_i simultaneously characterizes two meanings: how a user i is trusted (or followed) by others, and how the same user rates items. Again, the item-specific latent feature vector V_j depicts how an item j is rated by users. Together, the $W_i^T V_j$ indicates how other users follow user i to rate item j, which is the approximation of real score R_{ij} as well. Similarly, one can learn the feature matrices W, V and B simultaneously by minimizing the following objective:

$$\begin{aligned}\mathcal{L} = &\sum_{(i,j)\in\Omega} (g(W_i^{\mathrm{T}} V_j) - R_{ij})^2 + \sum_{(k,i)\in\Psi} (g(B_k^{\mathrm{T}} W_i) \\ &- T_{ki})^2 + \lambda(\sum_i (n_{w_i} + m_{w_i})\|W_i\|_{\mathrm{F}}^2 \\ &+ \sum_j n_{v_j}\|V_j\|_{\mathrm{F}}^2 + \sum_k m_{b_k}\|B_k\|_{\mathrm{F}}^2)\end{aligned} \tag{5}$$

where n_{w_i} and n_{v_j} denote the number of ratings given by user i and given to item j, respectively, m_{w_i} is the number of users who trust user i and m_{b_k} is the number of users who are trusted by user k.

2.3.3 Synthetic Influence of Trust Propagation

Individuals will affect each other during the process of reviewing and making decisions. This point is emphasized in our work. Therefore, your opinions/attitudes on items (e.g., whether purchase them or not, how to rate them, etc.) will be more or less affected by your trustees. Meanwhile, your decisions will inevitably influence the choices of your trusters. Comprehensively, it suggests that the observed ratings are actually generated according to the propagation of such twofold influence among users.

After independently training the truster and trustee models, one can obtain two sets of feature matrices. Let B_i^r and V_j^r be the truster-specific vector and item-specific vector learned by truster model. Let W_i^e and V_j^e be the trustee-specific vector and item-specific vector learned by trustee model. We suggest following synthetic strategy, named as TrustMF, to generate the approximation of real rating scores.

$$\widehat{R}_{ij} = g(\frac{(B_i^r)^{\mathrm{T}} V_j^r + (W_i^e)^{\mathrm{T}} V_j^e}{2}) \cdot R_{\max} \tag{6}$$

2.4 Insight into TrustMF Model

The key idea behind the proposed truster and trustee models is to construct a bridge between ratings and trust by mapping them into the same d-dimensional latent space, in which each dimension characterizes one of d aspects of items or users. Then we have, V_j (item feature), U_i (user feature), B_i (truster propensity) and W_i (trustee propensity) depict "what type item j is", "what type of items user i prefers", "in what aspects user i is willing to believe others", and "in what aspects user i deserves to be trusted", respectively. For Epinions as an example, B_i and W_i can also be explained as "what types of opinions (ratings/reviews/comments) user i prefers to browse" and "what types of opinions he prefers to generate", because browing/generating behaviors exactly

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2605085, IEEE Transactions on Pattern Analysis and Machine Intelligence

model how users follow/affect others. Note that one tends to give higher ratings to his preferred items and trust those whose opinions are relevant or helpful to him. Therefore, observed ratings and trust can be approximated by measuring the coincidence of respective feature vectors in terms of inner products, i.e., $R_{ij} \approx U_i^{\mathrm{T}} V_j$ and $T_{ik} \approx B_i^{\mathrm{T}} W_k$.

Our models allow recommendation systems to learn users' interest (preference) from their social activities. Specifically, the truster model learns users' interest from their browsing behaviors, in which U_i is approximated by B_i, implying that one will pay more attention to the opinions of items he prefers. On the other hand, the trustee model learns users' preference from their writing behaviors, in which U_i is approximated by W_i, implying that one will more likely generate opinions about items he prefers. As a result, users are encouraged to interact with the system, such that more social activities will result in more accurate recommendations for them. Truster and trustee models favor those users having more browsing and writing activities, respectively. However, TrustMF tries to provide high-quality recommendations for both kinds of users.

Note, the Truster model proposed here is similar to the SoRec model [21] simply in terms of mathematical form, both factorizing trust matrix T into two feature matrices. However, the basic idea, the motivation and the implication behind Truster are quite different from SoRec. SoRec is starting from the factorization of rating matrix $R \approx U^{\mathrm{T}} V$ (where U and V denote user and item feature matrices, respectively) and factorizing a trust network T into U and a factor feature matrix Z, in order to construct a bridge between R and T. Notably, as mentioned in a recent work [22], the main disadvantage of SoRec is that the factorization of T into U and Z "lacks physical interpretations, which does not reflect the real world recommendation process." In contrast, Truster model does not suffer from this problem. Truster model begins with analyzing the real-world online reviewing process and imitating the actual generation of trust relationships among users, by factorizing trust network T into B and W. As mentioned above, B and W depict the browsing and writing interesting of users, respectively, both reflecting the genuine interesting of users. By admitting this point, one can come up with a new model to enhance the learning performance of user preference, via simultaneously taking B and W into account when unifying the factorizations of ratings and trust relationships. It is clearly that, compared with $T \approx U^{\mathrm{T}} Z$ in SoRec, the factorization of $T \approx B^{\mathrm{T}} W$ in Truster or Trustee models provides more explicit and reasonable physical interpretations.

3 The TrustPMF Model

In Section 2, we proposed the Truster and Trustee models in which user feature vector U_i (genuine interest pattern) is deterministically replaced by the truster feature vector B_i (browsing interest pattern) or trustee feature vector W_i (writing interest pattern). This does not conform to our original intention very well, which tries to make U_i close to (not strictly equal to) B_i or W_i in modeling. In fact, a user's browsing interest or writing interest is not usually strictly identical to his/her genuine interest, such that there might be some fluctuations between U_i and B_i (W_i). Consequently, we further present a more general and flexible model named TrustPMF, which provides a probabilistic interpretation to the deterministic Truster/Trustee/TrustMF models, and can more accurately infer interest patterns of users.

3.1 Probabilistic Models of Truster and Trustee

In social collaborative filtering by trust, we have a rating matrix $R = [R_{ij}]_{m \times n}$, a trust matrix $T = [T_{ik}]_{m \times m}$, a user feature matrix $U \in \mathbb{R}^{d \times m}$, an item feature matrix $V \in \mathbb{R}^{d \times n}$, a truster feature matrix $B \in \mathbb{R}^{d \times m}$, and a trustee feature matrix $W \in \mathbb{R}^{d \times m}$.

The conditional distribution of observed ratings has been previously modeled as [29]:

$$p(R|U, V, \sigma^2) = \prod_{(i,j) \in \Omega} \mathcal{N}(R_{ij} | g(U_i^{\mathrm{T}} V_j), \sigma_R^2) \tag{7}$$

where $\mathcal{N}(x|\mu, \sigma^2)$ is the probability density function of the normal distribution with mean μ and variance σ^2, and $g(x) = 1/(1 + \exp(-x))$ is the logistic function. The priors of the user and item features are modeled as zero-mean spherical Gaussian distributions [7]:

$$\begin{aligned} p(U|\sigma_U^2) &= \prod_{i=1}^{m} \mathcal{N}(U_i | 0, \sigma_U^2 \mathbf{I}) \\ p(V|\sigma_V^2) &= \prod_{j=1}^{n} \mathcal{N}(V_j | 0, \sigma_V^2 \mathbf{I}) \end{aligned} \tag{8}$$

Similarly, we can model the observed trust relations, truster and trustee features by defining following probability distributions:

$$\begin{aligned} p(T|B, W, \sigma^2) &= \prod_{(i,k) \in \Psi} \mathcal{N}(T_{ik} | g(B_i^{\mathrm{T}} W_k), \sigma_T^2) \\ p(B|\sigma_B^2) &= \prod_{i=1}^{m} \mathcal{N}(B_i | 0, \sigma_B^2 \mathbf{I}) \\ p(W|\sigma_W^2) &= \prod_{k=1}^{m} \mathcal{N}(W_k | 0, \sigma_W^2 \mathbf{I}) \end{aligned} \tag{9}$$

To simplify the model, we assume the observed ratings and trust ties have identical noise levels, and use the same notation σ^2 to represent σ_R^2 and σ_T^2 in the rest of this paper. Similarly, we use the same notation σ_F^2 to represent the Gaussian prior variance of user (U), item (V), truster (B) and trustee (W) features. That is, we assume $\sigma_F = \sigma_U = \sigma_V = \sigma_B = \sigma_W$.

3.1.1 Truster-PMF

The Truster-PMF model is proposed to incorporate the users' browsing behaviors to infer their preferences, in which user feature vector U_i is approximated to truster feature vector B_i. In addition to the Gaussian prior $p(U|\sigma_F^2)$, we define a conditional distribution of U given B to model the influence of browsing behaviors on preference formation:

$$p(U|B, \sigma_M^2) = \prod_{i=1}^{m} \mathcal{N}(U_i | B_i, \sigma_M^2 \mathbf{I}) \tag{10}$$

where variance σ_M^2 controls the extent by which U_i approximates to B_i. Then, a new prior distribution over user feature vectors can be defined as:

$$p(U|B, \sigma_F^2, \sigma_M^2) \propto p(U|\sigma_F^2) p(U|B, \sigma_M^2) \tag{11}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2605085, IEEE Transactions on Pattern Analysis and Machine Intelligence

Through a Bayesian inference, the posterior distribution over the features of user, item, truster and trustee can be deduced as:

$$\begin{aligned} &p(U,V,B,W|R,T,\sigma^2,\sigma_F^2,\sigma_M^2) \\ &\propto p(R|U,V,\sigma^2)p(T|B,W,\sigma^2)p(U|\sigma_F^2) \\ &\quad p(U|B,\sigma_M^2)p(V|\sigma_F^2)p(B|\sigma_F^2)p(W|\sigma_F^2) \end{aligned} \tag{12}$$

We can use a probabilistic graphical model (see Fig.3) to describe the conditional dependencies among the random variables involved in Truster-PMF. We can see that the user feature vector U_i is dependent on the truster feature vector B_i, which properly meets our previous heuristic knowledge that a user's preference will be affected by his/her browsing behavior.

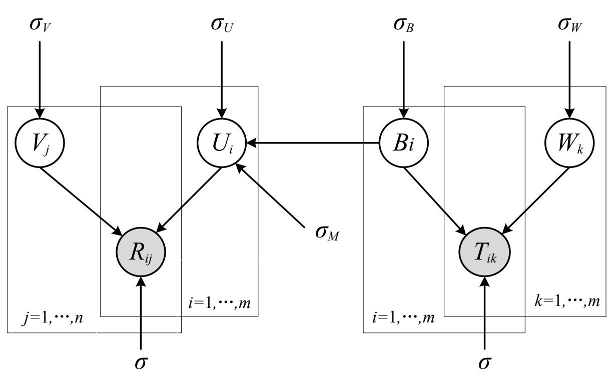


Fig. 3. The graphical model of Truster-PMF.

3.1.2 Trustee-PMF

The Trustee-PMF model is proposed to model the influence of writing behaviors on preference formation, in which the user feature vector U_i approximates to the trustee feature vector W_i. Similarly, we define a conditional distribution of U given W as follows:

$$p(U|W,\sigma_N^2) = \prod_{i=1}^{m} \mathcal{N}(U_i|W_i,\sigma_N^2\mathbf{I}) \tag{13}$$

where variance σ_N^2 controls the extent by which U_i approximates W_i. Then, a new prior distribution over user feature vectors can be defined as:

$$p(U|W,\sigma_F^2,\sigma_N^2) \propto p(U|\sigma_F^2)p(U|W,\sigma_N^2) \tag{14}$$

Correspondingly, the posterior distribution over the features of user, item, truster and trustee is given by:

$$\begin{aligned} &p(U,V,B,W|R,T,\sigma^2,\sigma_F^2,\sigma_N^2) \\ &\propto p(R|U,V,\sigma^2)p(T|B,W,\sigma^2)p(U|\sigma_F^2) \\ &\quad p(U|W,\sigma_N^2)p(V|\sigma_F^2)p(B|\sigma_F^2)p(W|\sigma_F^2) \end{aligned} \tag{15}$$

3.2 TrustPMF: Probabilistic Model of TrustMF

We now propose a synthetic probabilistic model to model the influence of both browsing and writing behaviors on preference formation, named as TrustPMF, by simultaneously considering truster features B and trustee features W. In this model, the prior distribution of user feature vectors is determined by three factors, a zero-mean Gaussian prior and two conditional distributions of U given B and W, defined as follows:

$$\begin{aligned} &p(U|B,W,\sigma_F^2,\sigma_M^2,\sigma_N^2) \\ &\propto p(U|\sigma_F^2)p(U|B,\sigma_M^2)p(U|W,\sigma_N^2) \end{aligned}$$

Correspondingly, the posterior distribution over the features of user, item, truster and trustee is given by:

$$\begin{aligned} &p(U,V,B,W|R,T,\sigma^2,\sigma_F^2,\sigma_M^2,\sigma_N^2) \\ &\propto p(R|U,V,\sigma^2)p(T|B,W,\sigma^2)p(U|B,\sigma_M^2) \\ &\quad p(U|W,\sigma_N^2)p(U|\sigma_F^2)p(V|\sigma_F^2)p(B|\sigma_F^2)p(W|\sigma_F^2) \end{aligned}$$

The graphical model of the TrustPMF is shown in Fig.4. In this model, user feature vector U_i is dependent on both truster feature vector B_i and trustee feature vector W_i, which means a user's preference shows not only in his/her browsing behavior but also in his/her writing behavior. In other words, TrustPMF model is able to infer users' preferences synthetically from item ratings, user browsing behaviors and user writing behaviors.

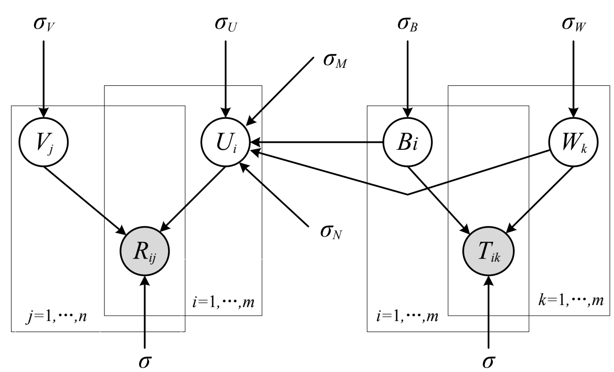


Fig. 4. The graphical model of TrustPMF.

To conveniently calculate the maximum posterior estimate of U, V, B and W, we get the log of above posterior probability distribution:

$$\begin{aligned} &\ln p(U,V,B,W|R,T,\sigma^2,\sigma_F^2,\sigma_M^2,\sigma_N^2) \\ &\propto -\frac{1}{2\sigma^2}\sum_{(i,j)\in\Omega}(R_{ij}-g(U_i^{\mathrm{T}}V_j))^2 - \frac{1}{2\sigma_F^2}\sum_{i=1}^{m}U_i^{\mathrm{T}}U_i \\ &- \frac{1}{2\sigma^2}\sum_{(i,k)\in\Psi}(T_{ik}-g(B_i^{\mathrm{T}}W_k))^2 - \frac{1}{2\sigma_F^2}\sum_{j=1}^{n}V_j^{\mathrm{T}}V_j \\ &- \frac{1}{2\sigma_M^2}\sum_{i=1}^{m}(U_i-B_i)^{\mathrm{T}}(U_i-B_i) - \frac{1}{2\sigma_F^2}\sum_{i=1}^{m}B_i^{\mathrm{T}}B_i \\ &- \frac{1}{2\sigma_N^2}\sum_{i=1}^{m}(U_i-W_i)^{\mathrm{T}}(U_i-W_i) - \frac{1}{2\sigma_F^2}\sum_{k=1}^{m}W_k^{\mathrm{T}}W_k \\ &- \frac{1}{2}(|\Omega|\ln\sigma^2 + |\Psi|\ln\sigma^2 + md\ln\sigma_M^2 + md\ln\sigma_N^2 \\ &+ md\ln\sigma_F^2 + nd\ln\sigma_F^2 + md\ln\sigma_F^2 + md\ln\sigma_F^2) + C \end{aligned}$$

where $|\Omega|$ denotes the number of observed ratings, $|\Psi|$ denotes the number of observed trust relations, and C is a constant that does not depend on the parameters. Maximizing the above log-posterior with the hyperparameters (i.e., the observation noise variance and prior variances) at fixed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2605085, IEEE Transactions on Pattern Analysis and Machine Intelligence

values is equivalent to minimizing the following objective function:

$$\mathcal{L} = \frac{1}{2}\sum_{(i,j)\in\Omega}(R_{ij} - g(U_i^{\mathrm{T}}V_j))^2 + \frac{\beta_1}{2}\sum_{i=1}^{m}\|U_i - B_i\|_{\mathrm{F}}^2$$
$$+ \frac{1}{2}\sum_{(i,k)\in\Psi}(T_{ik} - g(B_i^{\mathrm{T}}W_k))^2 + \frac{\beta_2}{2}\sum_{i=1}^{m}\|U_i - W_i\|_{\mathrm{F}}^2$$
$$+ \frac{\lambda}{2}(\sum_{i=1}^{m}\|U_i\|_{\mathrm{F}}^2 + \sum_{j=1}^{n}\|V_j\|_{\mathrm{F}}^2 + \sum_{i=1}^{m}\|B_i\|_{\mathrm{F}}^2 + \sum_{k=1}^{m}\|W_k\|_{\mathrm{F}}^2)$$

where $\beta_1 = \sigma^2/\sigma_M^2$, $\beta_2 = \sigma^2/\sigma_N^2$ and $\lambda = \sigma^2/\sigma_F^2$.

Note that, in previous models, the preferences of all users have a uniform variance, as shown in Eq.8, we suggest it is more reasonable to characterize different users with different prior variances for better-personalized recommendations. If we let n_{u_i} be the number of ratings given by user i, then one can adjust user i's prior variance according to n_{u_i}. The more ratings user i contributes, the better his/her preference can be estimated, and consequently, the smaller the uncertainty. This means that the prior variance of user i will be inversely proportional to n_{u_i}. Accordingly, the Gaussian prior over user features (U) is redefined as:

$$p(U|\sigma_F^2) = \prod_{i=1}^{m}\mathcal{N}(U_i|0, \frac{\sigma_F^2}{n_{u_i}}\mathbf{I}) \tag{16}$$

The same method can be used to redefine $p(V|\sigma_F^2)$, $p(B|\sigma_F^2)$ and $p(W|\sigma_F^2)$. As a result, we obtain the final objective of TrustPMF, as follows:

$$\mathcal{L} = \mathcal{X} + \frac{\beta_1}{2}\sum_{i=1}^{m}\|U_i - B_i\|_{\mathrm{F}}^2 + \frac{\beta_2}{2}\sum_{i=1}^{m}\|U_i - W_i\|_{\mathrm{F}}^2 \tag{17}$$

where

$$\mathcal{X} = \frac{1}{2}\sum_{(i,j)\in\Omega}(R_{ij} - g(U_i^{\mathrm{T}}V_j))^2$$
$$+ \frac{1}{2}\sum_{(i,k)\in\Psi}(T_{ik} - g(B_i^{\mathrm{T}}W_k))^2 + \frac{\lambda}{2}(\sum_{i=1}^{m}n_{u_i}\|U_i\|_{\mathrm{F}}^2$$
$$+ \sum_{j=1}^{n}n_{v_j}\|V_j\|_{\mathrm{F}}^2 + \sum_{i=1}^{m}m_{b_i}\|B_i\|_{\mathrm{F}}^2 + \sum_{k=1}^{m}m_{w_k}\|W_k\|_{\mathrm{F}}^2)$$

Parameter n_{v_j} denotes the number of ratings given to item j, m_{b_i} denotes the number of users who are trusted by user i, and m_{w_k} denotes the number of users who trust user k.

Similarly, we can obtain the objectives of Truster-PMF and Trustee-PMF based on Eq.12 and Eq.15, which are given in Eq.18 and Eq.19, respectively.

$$\mathcal{L}_{er} = \mathcal{X} + \frac{\beta}{2}\sum_{i=1}^{m}\|U_i - B_i\|_{\mathrm{F}}^2 \tag{18}$$

$$\mathcal{L}_{ee} = \mathcal{X} + \frac{\beta}{2}\sum_{i=1}^{m}\|U_i - W_i\|_{\mathrm{F}}^2 \tag{19}$$

Note, if we set $\beta_2 = 0$, TrustPMF (Eq.17) will turn into Truster-PMF (Eq.18), and if we set $\beta_1 = 0$, TrustPMF will turn into Trustee-PMF (Eq.19). In particular, when fix $\beta_2 = 0$ and set β_1 to a very large number, TrustPMF will theoretically be equivalent to the Truster model (see Eq.4). When fix $\beta_1 = 0$ and set β_2 to a very large number, TrustPMF will theoretically be equivalent to the Trustee model (see Eq.5). All above discussions indicate that the Truster-PMF, Trustee-PMF, Truster or Trustee are just special cases of the TrustPM-F. In other words, we justify the TrustMF model proposed in Section 2 from a probabilistic perspective. TrustPMF is a more general model, which can be deemed as a basic framework for social collaborative filtering by trust.

3.3 Training TrustPMF

Gradient descent algorithms can be used to train the aforementioned models, i.e., to minimize corresponding objective functions. Taking TrustPMF as an example, the gradients of $\mathcal{L}$ (in Eq.17) with respect to U_i, V_j, B_i and W_k are presented as follows, respectively:

$$\frac{\partial\mathcal{L}}{\partial U_i} = \sum_{j\in\mathcal{R}(i)}g'(U_i^{\mathrm{T}}V_j)(g(U_i^{\mathrm{T}}V_j) - R_{ij})V_j$$
$$+ \beta_1(U_i - B_i) + \beta_2(U_i - W_i) + \lambda n_{u_i}U_i,$$
$$\frac{\partial\mathcal{L}}{\partial V_j} = \sum_{i\in\mathcal{R}^+(j)}g'(U_i^{\mathrm{T}}V_j)(g(U_i^{\mathrm{T}}V_j) - R_{ij})U_i$$
$$+ \lambda n_{v_j}V_j,$$
$$\frac{\partial\mathcal{L}}{\partial B_i} = \sum_{k\in\mathcal{F}(i)}g'(B_i^{\mathrm{T}}W_k)(g(B_i^{\mathrm{T}}W_k) - T_{ik})W_k$$
$$- \beta_1(U_i - B_i) + \lambda m_{b_i}B_i,$$
$$\frac{\partial\mathcal{L}}{\partial W_k} = \sum_{i\in\mathcal{F}^+(k)}g'(B_i^{\mathrm{T}}W_k)(g(B_i^{\mathrm{T}}W_k) - T_{ik})B_i$$
$$- \beta_2(U_k - W_k) + \lambda m_{w_k}W_k$$

where $\mathcal{R}(i)$ denotes the set of items that user i has rated, $\mathcal{R}^+(j)$ denotes the set of users who have rated item j, $\mathcal{F}(i)$ denotes the set of users who are trusted by user i, and $\mathcal{F}^+(k)$ denotes the set of users who trust user k. $g'(x) = \exp(-x)/(1+\exp(-x))^2$ is the derivative of the logistic function $g(x)$.

In order to more conveniently learn the parameters of the TrustPMF, we choose the function $f(x) = x/(1 + R_{\max})$ to map the raw rating R_{ij} into the interval $(0, 1)$. Correspondingly, we also transform the trust value T_{ik} by employing a function $f(y) = (R_{\max}/(1+R_{\max}))y$ to ensure the maximum of T_{ik} is equal to the maximum of R_{ij} during the process of training. The final prediction is generated by $g(U_i^{\mathrm{T}}V_j)\cdot(1 + R_{\max})$ after model training. The gradient descent algorithm for TrustPMF is presented in Table 1.

The time of training TrustPMF is mainly taken by computing the objective function $\mathcal{L}$ and its gradients against feature vectors. The computing time of $\mathcal{L}$ is $O(td(|\Omega|+|\Psi|))$, where t is the required iterations, d is the dimensionality of feature vectors, $|\Omega|$ and $|\Psi|$ are the numbers of observed ratings and observed trust links, respectively. The costs of computing the gradients of $\mathcal{L}$ against U, V, B and W are $O(td|\Omega|)$, $O(td|\Omega|)$, $O(td|\Psi|)$ and $O(td|\Psi|)$, respectively. Therefore, the time complexity of the TrustPMF training algorithm is $O(td(|\Omega|+|\Psi|))$, linearly scaling to the numbers of observed ratings and trust links.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2605085, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 1
The Training Algorithm of TrustPMF

Algorithm 1 TrustPMF

Input: $R, T, d, \lambda, \beta_1, \beta_2, \alpha$(learning rate)

1. Initialize U, V, B, W with small random numbers
2. **while** $\mathcal{L}$ has not converged:
3. Compute $\frac{\partial \mathcal{L}}{\partial U_i}, \frac{\partial \mathcal{L}}{\partial V_j}, \frac{\partial \mathcal{L}}{\partial B_i}$ and $\frac{\partial \mathcal{L}}{\partial W_k}$ for all users or items
4. $U_i \leftarrow U_i - \alpha \cdot \frac{\partial \mathcal{L}}{\partial U_i}, \quad i = 1, ..., m$
5. $V_j \leftarrow V_j - \alpha \cdot \frac{\partial \mathcal{L}}{\partial V_j}, \quad j = 1, ..., n$
6. $B_i \leftarrow B_i - \alpha \cdot \frac{\partial \mathcal{L}}{\partial B_i}, \quad i = 1, ..., m$
7. $W_k \leftarrow W_k - \alpha \cdot \frac{\partial \mathcal{L}}{\partial W_k}, \quad k = 1, ..., m$
8. **end while**
9. $r_{ij} \leftarrow g(U_i^{\mathrm{T}} V_j) \cdot (1 + R_{\max})$, where $r_{ij} \in [0, 1 + R_{\max}]$

Output: The predicted rating $\widehat{R}_{ij} \leftarrow \begin{cases} 1, & r_{ij} < 1 \\ R_{\max}, & r_{ij} > R_{\max} \\ r_{ij}, & \text{otherwise} \end{cases}$

4 Experiments and Validations

4.1 Description of Datasets

In order to sufficiently validate the performance of our proposed methods, we choose four representative data sets related to social collaborative filtering for our experiments, which are taken from popular social networking web sites including Epinions (www.epinions.com), Douban (www.douban.com) and Flixster (www.flixster.com). Each of these social networking services allows users to rate items, browse/write reviews, add friends, and hence provide ratings, reviews and social relations. Note, the trust network in Epinions is directed, while the friend networks in Douban and Flixster are undirected because the new friend requests in these two web sites must be verified and approved by both parties. Therefore, a friend network can be deemed as a special trust network, in which friends trust each other. In all datasets, rating scales from 1 to 5. The statistics of the four datasets are presented in Table 2.

TABLE 2
Statistics of Datasets

Statistics	Epinions	Ep_Ext	Douban	Flixster
#Users	49,289	120,492	129,490	147,612
#Items	139,738	755,760	58,541	48,794
#Ratings (Density)	664,823 0.0097%	13,668,320 0.0150%	16,830,939 0.2220%	8,196,077 0.1138%
#Relations (Density)	487,183 0.0201%	606,259 0.0042%	1,711,780 0.0102%	2,538,746 0.0117%
#Ratings per User	13.49	113.44	129.98	55.52
#Ratings per Item	4.76	18.09	287.51	167.97
#Friends per User	9.88	5.03	13.22	17.20

Epinions & Ep_Ext Both two datasets are published by [24]. Epinions is extremely sparse, containing 49,289 users, 139,738 items, 664,823 ratings and 487,183 trust relations. The items in Ep_Ext refer to articles (i.e., reviews) written by users rather than typical products, but can still be used for validating CF methods. Ep_Ext also contains distrust relations, which we do not consider in this work, and are removed. We also removed users or items with no ratings because they are meaningless to experimental evaluation. The Ep_Ext we finally obtained contains 120,492 users, 755,760 items, 13,668,320 ratings and 606,259 trust relations.

Douban This dataset is published by the authors of [23], which contains 129,490 users, 58,541 items, 16,830,839 ratings and 1,692,952 friend relations. However, it missed some mutual relations that should appear in the dataset (e.g., (a, b) exists in the dataset while (b, a) do not appear in it). We supplement those missed relations into the dataset and remove 22 "reflexive" relations (i.e., (a, a)-like relations) in the original dataset. In total, there are 1,711,780 social relations in the dataset.

Flixster This dataset is published by [14]. The original dataset contains 1,049,511 users, 66,726 items, 8,196,077 ratings and 11,794,648 social relations. In this dataset, most of users have not rated any items. We removed those users or items with no ratings, which are meaningless to experimental evaluation. Finally, the dataset we obtained contains 147,612 users, 48,794 items, 8,196,077 ratings and 2,538,746 social relations.

4.2 Experimental Setup

4.2.1 Comparison Methods

To comparatively evaluate the performance of our proposed methods, we select four representative CF methods as competitors: one traditional matrix factorization based method PMF [29] that does not consider social ties among users, and three recently proposed social recommendation methods, SoRec [21], RSTE [22] and SocialMF [14], which can efficiently handle the aforementioned large-scale datasets. One can find the introductions about them in Section 1.

To focus on verifying models and fair comparison, for all methods, we set the same dimensionality of latent space $d = 10$, the same regularization parameter $\lambda = 0.001$, and adopt the same initialization strategy, which randomly initializes all involved feature matrices using a Gaussian distribution with a mean of 0 and a variance of 0.01. In addition, for each method, we set the respective optimal parameters according to their corresponding references or based on our experiments, given in Table 3.

TABLE 3
Parameter Settings of Compared Methods

Methods	Optimal parameters	
PMF [29]	λ_U=λ_V=0.001	
SoRec [21]	λ_U=λ_V=λ_Z=0.001, λ_C=1	
RSTE [22]	λ_U=λ_V=0.001, α=0.4	
SocialMF [14]	λ_U=λ_V=0.001, λ_T=1	
TrustMF	λ=0.001	
Truster-PMF Trustee-PMF	**Epinions**	λ=0.001, β=10
	Ep_Ext	λ=0.001, β=1
	Douban	λ=0.001, β=0.5
	Flixster	λ=0.001, β=1
TrustPMF	**Epinions**	λ=0.001, β_1=β_2=10
	Ep_Ext	λ=0.001, β_1=β_2=0.5
	Douban	λ=0.001, β_1=β_2=0.5
	Flixster	λ=0.001, β_1=β_2=0.2

4.2.2 Cross-validation

We employ cross-validation for learning and testing. For the experiments conducted on datasets Ep_Ext, Douban

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2605085, IEEE Transactions on Pattern Analysis and Machine Intelligence

and Flixster, we use a 2-fold cross-validation. We randomly divide the rating data into two equal parts; in each time we use one part as training set (50% of data) and another part as test set (the remaining 50% of data). Since Epinions is extremely sparse in terms of ratings, we select a larger part of the data for training. For this dataset, we use a 5-fold cross-validation that randomly divides the rating data into five equal parts, in each time we use one part as test set (20% of data) and the remaining four parts as the training set (the remaining 80% of data). In addition, we conduct each experiment five times and then take the mean as the final result for each experiment discussed below.

4.3 Experimental Results

4.3.1 Validation on All Users

We now validate the performance of the proposed TrustMF and TrustPMF methods by comparing the results on the four datasets with their competitors. MAE (mean absolute error) and RMSE (root mean square error), two benchmark error evaluation metrics, are used here. Table 4 shows the results of MAE and RMSE computed on the predictions for all users, where Tr-PMF, Te-PMF and TPMF refer to Truster-PMF, Trustee-PMF and TrustPMF, respectively. From the table, we can see the following points.

First, TrustPMF (sometimes Truster-PMF or Trustee-PMF) performs the best of all in terms of both MAE and RMSE on all four datasets. The improvements against respective competitors given in the table show that our methods can significantly improve the quality of recommendations, especially for Epinions, a highly sparse dataset.

Second, TrustMF achieves a second-best performance except on the dataset Ep_Ext in terms of MAE.

Third, because the rating data of Epinions is extremely sparse (density of 0.0097%), the traditional CF method PMF performs much worse than the social recommendation methods SoRec, RSTE and SocialMF. However, when the rating data is relatively dense, such as in the Douban (density of 0.2220%) or Flixster (density of 0.1138%), PMF shows a comparable, and sometimes better, performance than them.

Finally, for Epinions and Ep_Ext that both contain directed trust networks, TrustPMF is more accurate than Truster-PMF or Trustee-PMF. However, for Douban and Flixster that contain undirected friend networks, the performance of Truster-PMF and Trustee-PMF is similar to TrustPMF. This is mainly because the browsing interest pattern (B) and writing interest pattern (W) learned from the symmetric friend networks are nearly identical, and hence the recommendation quality improvement that results from their combination is limited.

In summary, the last two rows of Table 4 give the average ranks of all methods in terms of both MAE and RMSE, reflecting their respective comprehensive performance on different datasets.

4.3.2 Validation on Cold Start Users

As mentioned in Section 1, cold start is one big challenge faced by CF methods. We now evaluate the capabilities of addressing cold start users by respective competitors. Generally, those who have rated five or fewer ratings are deemed as cold start users [13], [14], [25]. Cross-validation is still used in the test but we only care about the accuracy of predictions for cold start users (with five or fewer ratings in training set). The number (average over all training sets) of cold start users in Epinions, Ep_Ext, Douban and Flixster are 30,166 (61.20%), 88,102 (73.12%), 17,659 (13.64%) and 100,296 (67.95%), respectively. And the social relation number (outboud plus inbound) on average of cold start user in Epinions, Ep_Ext, Douban and Flixster are 5.38, 1.47, 23.97 and 27.43, respectively.

Table 5 shows that TrustPMF performs the best once again on datasets Epinions, Ep_Ext and Flixster while TrustMF performs the best on Douban, and achieve more remarkable improvements against other CF methods than the case of testing all users. This experiment indicates that, compared with current CF methods, our proposed TrustPMF and TrustMF have demonstrated the ability to provide high quality recommendations to cold start users.

4.3.3 Validation on Users with Different Degrees

We conduct an additional experiment to further check the capabilities of the different methods to utilize sparse trust (social relation) data for recommendation. Distinct from previous validations that focus on comparing the global quality of recommendations in terms of the average accuracy over all users, here we are particularly interested in testing the performance of respective social CF methods (we still take PMF as baseline method) with regard to different categories of users. We first cluster all users into several groups according to their degrees (outdegree plus indegree) in social networks, and then calculate the prediction error over each group. The degree distribution of each group of users and the average number of ratings in each group are shown in Fig. 5. As we see, for each dataset, the more social ties uses have, the more ratings they will contribute.

Fig. 6 shows the recommendation quality of different methods in terms of RMSE (similar trends for MAE), from which we can observe following points.

First, TrustPMF demonstrates the highest quality for all groups on all datasets, particularly for users who occupy only a few trust ties, indicating that TrustPMF can utilize sparse trust data more effectively than current social CF methods.

Second, the performance of all six methods varies to different extents with respect to different groups, but the result curves of the different methods have similar trends.

Moreover, for all datasets except Ep_Ext, and for all social CF methods, as well as the traditional CF method PMF, the groups with higher degrees (groups "101∼200" and ">200") do not achieve higher prediction accuracy, which is contrary to our expectation. This means that users who occupy more social ties will contribute more to ratings, and consequently the predictions for them are expected to be more accurate. This indicates that some users with more social relations (even those have more ratings) might not receive desired recommendations from recommender systems. One possible reason for this is that some active users tend to add more and more friends, or trust more and more users, without real intentions, and these casual social ties will greatly reduce the ability to learn their genuine interest patterns.

TABLE 4
Experimental Results on Testing All Users

Datasets	Metrics	PMF	SoRec	RSTE	SocialMF	TrustMF	Tr-PMF	Te-PMF	TPMF
Epinions	MAE	1.1124	1.0401	1.0281	1.0366	1.0142	0.9666	0.9548	**0.9396**
	(Improve)	15.53%	9.66%	8.61%	9.36%	7.36%	2.79%	1.59%	–
	RMSE	1.3528	1.2757	1.3290	1.3136	1.2725	1.1986	1.1860	**1.1674**
	(Improve)	13.70%	8.49%	12.16%	11.13%	8.26%	2.60%	1.57%	–
Ep_Ext	MAE	0.3311	0.3296	0.3212	0.3267	0.3850	0.3166	0.3179	**0.3112**
	(Improve)	6.01%	5.58%	3.11%	4.74%	19.17%	1.71%	2.11%	–
	RMSE	0.5856	0.5797	0.6047	0.5702	0.5563	0.5308	0.5337	**0.5306**
	(Improve)	9.39%	8.47%	12.25%	6.94%	4.62%	0.04%	0.58%	–
Douban	MAE	0.5789	0.5793	0.5799	0.5824	0.5684	**0.5639**	**0.5639**	0.5651
	(Improve)	2.59%	2.66%	2.76%	3.18%	0.79%	–	–	0.21%
	RMSE	0.7344	0.7333	0.7332	0.7366	0.7133	0.7112	0.7113	**0.7109**
	(Improve)	3.20%	3.05%	3.04%	3.49%	0.34%	0.04%	0.06%	–
Flixster	MAE	0.6595	0.6643	0.6584	0.6590	0.6545	0.6465	**0.6462**	**0.6462**
	(Improve)	2.02%	2.72%	1.85%	1.94%	1.27%	0.05%	–	–
	RMSE	0.8921	0.8931	0.8863	0.8839	0.8643	0.8617	0.8614	**0.8612**
	(Improve)	3.46%	3.57%	2.83%	2.57%	0.36%	0.06%	0.02%	–
Avg. Rank	MAE	6.25	6.25	4.75	5.75	4.5	2	1.75	**1.25**
	RMSE	7.25	6.25	6.5	6	4	2.5	2.5	**1**

TABLE 5
Experimental Results on Testing Cold Start Users

Datasets	Metrics	PMF	SoRec	RSTE	SocialMF	TrustMF	Tr-PMF	Te-PMF	TPMF
Epinions	MAE	1.3628	1.2544	1.2995	1.2467	1.2211	1.1573	1.1003	**1.0411**
	(Improve)	23.61%	17.00%	19.88%	16.49%	14.74%	10.04%	5.38%	–
	RMSE	1.5205	1.4257	1.5395	1.4906	1.4162	1.3368	1.2771	**1.2300**
	(Improve)	19.11%	13.73%	20.10%	17.48%	13.15%	7.99%	3.69%	–
Ep_Ext	MAE	1.4152	1.3669	1.3645	1.2471	1.1114	1.0870	1.1245	**1.0803**
	(Improve)	23.66%	20.97%	20.83%	13.38%	2.80%	0.62%	3.93%	–
	RMSE	1.5380	1.4940	1.6293	1.5411	1.3700	1.2774	1.3101	**1.2760**
	(Improve)	17.04%	14.59%	21.68%	17.20%	6.86%	0.11%	2.60%	–
Douban	MAE	0.8365	0.7611	0.7759	0.7409	**0.7234**	0.7472	0.7482	0.7461
	(Improve)	13.52%	4.95%	6.77%	2.36%	–	3.19%	3.31%	3.04%
	RMSE	1.0260	0.9371	0.9636	0.9221	**0.8960**	0.9052	0.9065	0.9026
	(Improve)	12.67%	4.39%	7.02%	2.83%	–	1.02%	1.16%	0.73%
Flixster	MAE	0.9864	0.9539	0.9608	0.9156	0.8943	0.8880	0.8889	**0.8810**
	(Improve)	10.69%	7.64%	8.31%	3.78%	1.49%	0.79%	0.89%	–
	RMSE	1.2369	1.2063	1.2197	1.1628	1.1207	1.1038	1.1059	**1.1001**
	(Improve)	11.06%	8.80%	9.81%	5.39%	1.84%	0.34%	0.52%	–
Avg. Rank	MAE	8	6.25	6.75	4.25	3	2.75	3.5	**1.5**
	RMSE	7.25	5.5	7.5	5.75	3.25	2.5	3	**1.25**

This experiment demonstrates that social networks can be used to improve the performance of collaborative filtering, and have potential to help recommendation systems target "extremely social" users for whom it is hard to provide accurate recommendations.

4.3.4 Validation on rank-based metrics

Now we use the metrics that are widely used in IR (information retrieval) to further evaluate different algorithms. Specifically, we use Precision, Recall and F1-score to test if algorithms can precisely and completely find all favorite items for target user, and use NDCG (Normalized Discounted Cumulative Gain) to quantitatively measure the consistency between the item rankings generated by algorithms and the ground truths.

Cross-validation is still used in the test. Let $Fav(i) = \{j \in \Omega(i) | R_{ij} \geq 4\}$ denotes the set of favorite items of user i, which is defined in terms of the ground truth R_{ij} in testing set. $\Omega(i)$ denotes the set of items that user i has rated in testing set. Remember, R_{ij} is an integer number within the interval [1,5], representing the extent how user i likes item j, e.g., "4" and "5" usually indicating user i favors item j. Let $Rec(i) = \{j \in \Omega(i) | r(\widehat{R}_{ij}) \geq 4\}$ denotes the set of recommended items to user i, which is defined based on the predictions of actual ratings. $r()$ denotes the rounding function, which rounds a real number $\widehat{R}_{ij}$ to an integer rating. m is the number of users. Then, Precision, Recall and F1-score are defined as follows:

$$\text{Precision} = \frac{1}{m}\sum_{i=1}^{m}\frac{|Rec(i) \bigcap Fav(i)|}{|Rec(i)|},$$

$$\text{Recall} = \frac{1}{m}\sum_{i=1}^{m}\frac{|Rec(i) \bigcap Fav(i)|}{|Fav(i)|},$$

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

NDCG is a rank-based metric widely used in IR. In the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2605085, IEEE Transactions on Pattern Analysis and Machine Intelligence

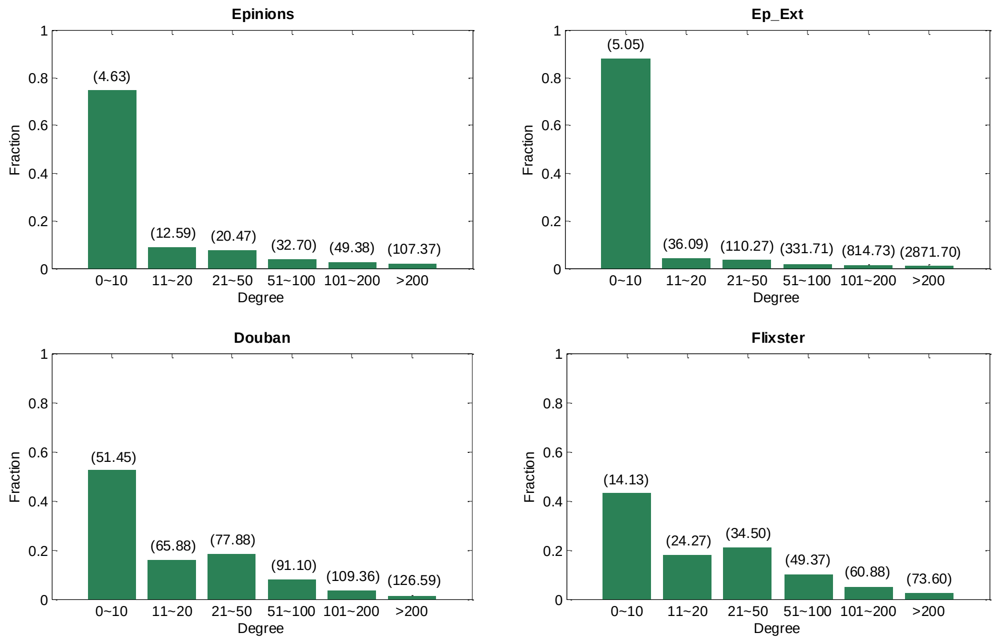


Fig. 5. Degree distribution of each group of users. The average number of ratings of each group is shown on the top of corresponding bar.

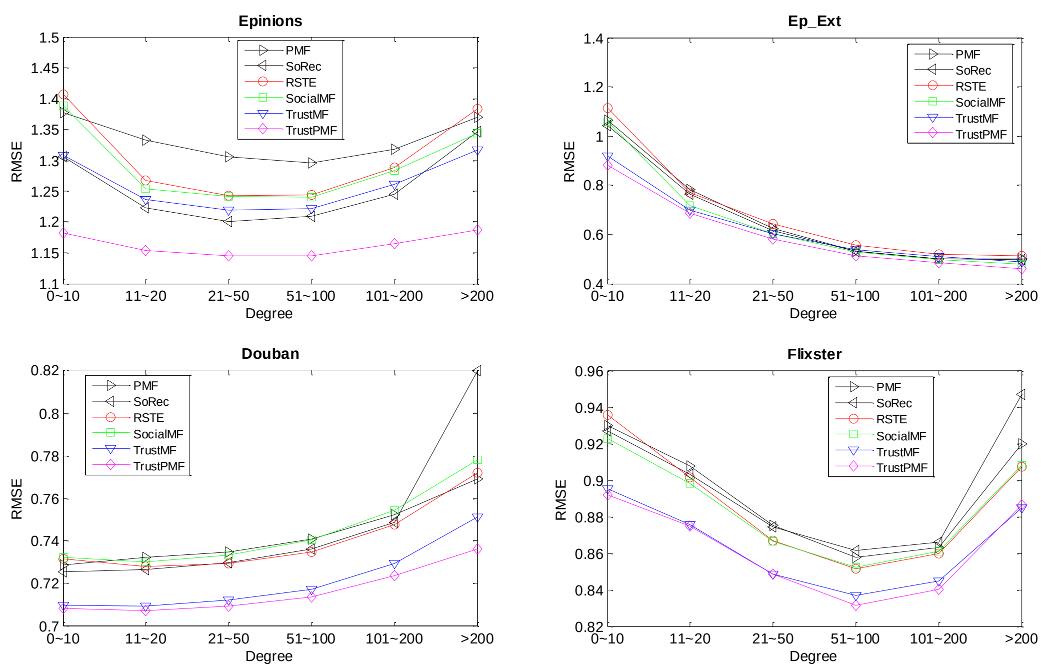


Fig. 6. Experimental results on testing different group of users.

case of recommendation, NDCG is defined as:

$$\text{NDCG} = \frac{1}{m}\sum_{i=1}^{m}\text{NDCG}(i) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{Z_i}DCG(i)$$

where

$$DCG(i) = \sum_{j\in\Omega(i)}\frac{2^{R_{ij}}-1}{\log_2(1+rank(j))}$$

$rank(j)$ denotes the position of item j in the sorted list of $\Omega(i)$, according to the predicted ratings $\widehat{R}_{ij}$. The normalized factor Z_i is the DCG value of the actual ranking of $\Omega(i)$, according to the ground truth R_{ij} in testing set. Note, NDCG value is between 0 and 1; the bigger the NDCG value, the better the ranking.

Based on the metrics, we tested the PMF, SoRec, RSTE, SocialMF and our proposed TrustPMF listed in Table 3 across four datasets in Table 2. Limited by space, here we just report the results of Epinions and Douban, two representative datasets, in the sense that Epinions contains a directed trust network while Douban contains a undirected friend network. Similar results are obtained for the other two datasets.

Tables 6 and 7 show the validations in terms of preci-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2605085, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 6
Performance of IR Metrics on Epinions Dataset

	Metrics	PMF	SoRec	RSTE	SocialMF	TrustPMF
All Users	Precision	0.5918	0.6693	0.5137	0.6685	**0.7789**
	(Improve)	31.62%	16.38%	51.63%	16.51%	–
	Recall	0.4985	0.5910	0.4244	0.5993	**0.7278**
	(Improve)	46.00%	23.15%	71.49%	21.44%	–
	F1-score	0.5412	0.6277	0.4648	0.6320	**0.7525**
	(Improve)	39.04%	19.88%	61.90%	19.07%	–
	NDCG	0.9255	0.9306	0.9245	0.9318	**0.9371**
	(Improve)	1.25%	0.70%	1.36%	0.57%	–
Cold-start Users	Precision	0.2988	0.4357	0.2750	0.4804	**0.7066**
	(Improve)	136.48%	62.18%	156.95%	47.09%	–
	Recall	0.2653	0.4016	0.2459	0.4590	**0.6754**
	(Improve)	154.58%	68.18%	174.66%	47.15%	–
	F1-score	0.2811	0.4180	0.2596	0.4695	**0.6906**
	(Improve)	145.68%	65.22%	166.02%	47.09%	–
	NDCG	0.9369	0.9431	0.9384	0.9449	**0.9532**
	(Improve)	1.74%	1.07%	1.58%	0.88%	–

TABLE 7
Performance of IR Metrics on Douban Dataset

	Metrics	PMF	SoRec	RSTE	SocialMF	TrustPMF
All Users	Precision	0.7735	0.7885	0.6768	0.7912	**0.7936**
	(Improve)	2.60%	0.65%	17.26%	0.30%	–
	Recall	0.8160	0.8535	0.6677	0.8565	**0.8788**
	(Improve)	7.70%	2.96%	31.62%	2.60%	–
	F1-score	0.7942	0.8197	0.6722	0.8225	**0.8336**
	(Improve)	4.96%	1.70%	24.01%	1.35%	–
	NDCG	0.9323	0.9273	0.9108	0.9347	**0.9388**
	(Improve)	0.70%	1.24%	3.07%	0.44%	–
Cold-start Users	Precision	0.6105	0.7707	0.5589	0.7873	**0.7967**
	(Improve)	30.50%	3.37%	42.55%	1.19%	–
	Recall	0.5711	0.7746	0.5380	0.7876	**0.8039**
	(Improve)	40.76%	3.78%	49.42%	2.07%	–
	F1-score	0.5902	0.7726	0.5483	0.7875	**0.7987**
	(Improve)	35.33%	3.38%	45.67%	1.42%	–
	NDCG	0.9451	0.9488	0.9396	0.9508	**0.9524**
	(Improve)	0.77%	0.38%	1.36%	0.17%	–

sion, recall and F1-score on all users and cold start users, respectively. On Epinions, TrustPMF performs much better than others in both cases. On Douban, TrustPMF performs better than SocialMF and SoRec and much better than PMF and RSTE in both cases.

Tables 6 and 7 also show the experimental results in terms of NDCG. In both cases of all users and cold start users, TrustPMF performs better than others, especially for Einions dataset. Notably, in terms of NDCG all methods perform better in the case of cold start users on both datasets. That is, the NDCG values calculated for cold start users are larger than those for all users. The main reasons are two-fold. On one hand, it is more difficult for algorithms to correctly rank a big set than a small one under the NDCG metric. On the other hand, cold start users usually occupy much less observed ratings in testing set and thereby the item sets to be ranked for them are very small.

Very recently, two new social recommendation methods, TrustSVD [11] and SPF (social Poisson factorization) [5], have been proposed. TrustSVD incorporates the ideas of both the SVD++ [16] and the Truster model [37], aiming to capture user and item biases as well as the explicit and implicit influence of both rated items and trusted users. SPF extends the Poisson factorization model [9] to integrate the implicit feedbacks of users and the social networks among the same users. Next, we will evaluate the performance of the two state-of-the-art methods and the TrustPMF in terms of rank-based metrics. These three model-based methods are also compared to a popularity baseline, which naively ranks items only according to their universal popularity in terms of the quantity of ratings received by them respectively.

In addition to the NDCG used above, four top-k based ranking metrics are also used in this experiment. They are: Precision@k (Pre@k for short), Recall@k (Rec@k for short), F1@k and MAP. These metrics highlight the significance of top-k recommendation and can help us to evaluate whether the top-k items recommended by algorithms are truly favored by users. Their definitions are given as follows:

$$\text{Pre@}k = \frac{1}{m}\sum_{i=1}^{m}\text{Pre@}k(i) = \frac{1}{m}\sum_{i=1}^{m}\frac{|\pi_k(i)\bigcap Fav(i)|}{k},$$

$$\text{Rec@}k = \frac{1}{m}\sum_{i=1}^{m}\text{Rec@}k(i) = \frac{1}{m}\sum_{i=1}^{m}\frac{|\pi_k(i)\bigcap Fav(i)|}{|Fav(i)|},$$

$$\text{F1@}k = \frac{2 \times \text{Pre@}k \times \text{Rec@}k}{\text{Pre@}k + \text{Rec@}k},$$

where Pre@$k(i)$ and Rec@$k(i)$ denote the values of Pre@k and Rec@k for user i, respectively, $Fav(i)$ is defined as above and denotes the set of favorite items of user i, and $\pi_k(i)$ denotes the set of the top k items in the ranking determined by algorithms for user i.

MAP (Mean Average Precision) evaluates the overall benefit of a ranking, which is defined in terms of Pre@k:

$$\text{MAP} = \frac{1}{m}\sum_{i=1}^{m}\frac{\sum_{k=1}^{s}\text{Pre@}k(i)\cdot I_{i,k}}{|Fav(i)|}$$

where $s = |\Omega(i)|$ is the length of the ranking list of user i, and $I_{i,k}$ is a binary indicator. $I_{i,k}$ is equal to 1 if the item at the k-th position is favored by user i and 0 otherwise.

The same experimental settings are used in this experiment. We set the common parameters $d = 10$ and $\lambda = 0.001$, and other parameters with default values according to the original papers. The experimental results regarding NDCG, MAP, Pre@k, Rec@k and F1@k (with k = 5,10) on the two representative datasets, Epinions and Douban, are shown in Table 8. Note that, the results of SPF on Douban are not available as it cannot converge within a reasonable time on this large dataset. It can be seen from Table 8 that TrustPMF outperforms TrustSVD on both datasets, and performs much better than SPF on Epinions.

5 Conclusions and Discussions

Data sparsity and cold starts present challenges to traditional collaborative filtering methods used in recommendation systems. We addressed these issues by more effectively utilizing additional trust data, and proposed a novel method based on social collaborative filtering by trust that was motivated by the observation that individuals will affect each other during the reviewing process. By properly identifying the twofold influence of trust propagation on the generation of a user's preferences, we first propose a truster model and a trustee model, and then combine them into the TrustMF

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2605085, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 8
Comparisons with TrustSVD, SPF and Popularity in Terms of Top-k Based Ranking Metrics

Datasets	Methods	NDCG	MAP	Pre@5	Rec@5	F1@5	Pre@10	Rec@10	F1@10
Epinions	Popularity	0.8108	0.8156	0.7265	0.6017	0.6583	0.7214	0.6321	0.6738
	SPF	0.8858	0.8216	0.7367	0.6066	0.6653	0.7298	0.6366	0.6800
	TrustSVD	0.9323	0.8925	0.8195	0.6644	0.7338	0.8081	0.6952	0.7474
	TrustPMF	**0.9371**	**0.8964**	**0.8279**	**0.6672**	**0.7389**	**0.8159**	**0.6985**	**0.7526**
Douban	Popularity	0.8979	0.7966	0.8247	0.2383	0.3697	0.7972	0.3285	0.4653
	TrustSVD	0.9353	0.8809	0.9092	0.2617	0.4064	0.8915	0.3626	0.5155
	TrustPMF	**0.9388**	**0.8853**	**0.9172**	**0.2633**	**0.4092**	**0.8975**	**0.3645**	**0.5184**

model. The truster and trustee models map users into the same latent feature spaces, but with different implications that can explicitly describe the feedback of how users affect or follow the opinions of others. Moreover, a more general and flexible framework referred to as the TrustPMF is proposed, which is able to provide a probabilistic view for understanding the Truster, Trustee and TrustMF models, and can further enhance recommendation quality by incorporating both browsing and writing preferences in a more flexible way. As shown by a series of vigorous validations, the TrustPMF performed significantly better than current CF methods across four large-scale datasets. This was especially true for the cold start users who have few ratings, or occupy only a few trust ties.

In addition to positive ratings (representing like) and positive social ties (representing trust), the available negative ratings (representing dislike) and negative social ties (representing distrust) are equally important for comprehensively learning a user's preferences. It is readily to extend the TrustPMF model to incorporate the influence of such negative factors on the formation of user preference and further improve the quality of social recommendations. For example, we have extended the basic TrustMF model to incorporate the distrust relationships among users [4]:

$$
\begin{aligned}
\mathcal{L} = & \sum_{(i,j)\in\Omega} (g(\beta B_i^{\mathrm{T}} V_j + (1-\beta) W_i^{\mathrm{T}} V_j) - R_{ij})^2 \\
& + \sum_{(i,k)\in\Psi} (g(B_i^{\mathrm{T}} W_k) - T_{ik})^2 \\
& - \lambda_1 \sum_{i=1}^{m} \sum_{j\epsilon D(i)} \|B_i - B_j\|^2 \\
& - \lambda_2 \sum_{i=1}^{m} \sum_{j\epsilon D(i)} \|W_i - W_j\|^2 \\
& + \lambda_b \|B\|_{\mathrm{F}}^2 + \lambda_w \|W\|_{\mathrm{F}}^2 + \lambda_v \|V\|_{\mathrm{F}}^2
\end{aligned}
$$

$D(i)$ denotes the set of users that i distrusts. The added penalty terms imply that the preference difference between two distrusted users should be as large as possible. It will be interesting to study more extensions in future work.

Acknowledgments

This work was funded by the National Natural Science Foundation of China under grants 61373053, 61133011, 61572226, 61300146, and 61272291, and Jilin Province Natural Science Foundation under grant 20150101052JC.

References

[1] G. Adomavicius, and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Trans. Knowledge and Data Eng.*, vol. 17, no. 6, pp. 734-749, 2005.

[2] P. Bedi, H. Kaur, and S. Marwaha, "Trust Based Recommender System for Semantic Web," *Proc. 20th Int'l Joint Conf. Artifical Intelligence (IJCAI'07)*, pp. 2677-2682, 2007.

[3] J.S. Breese, D. Heckerman, and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," *Proc. 14th Conf. Uncertainty in Artificial Intelligence (UAI'98)*, pp. 43-52, 1998.

[4] T. Bai, B. Yang, and F. Li, "TDRec: Enhancing Social Recommendation Using Both Trust and Distrust Information," *Proc. 2nd European Network Intelligence Conference*, pp. 60-66, 2015.

[5] A. J. B. Chaney, D. M. Blei, and T. Eliassi-Rad, "A Probabilistic Model for Using Social Networks in Personalized Item Recommendation," *Proc. 9th ACM Conf. Recommender Systems*, pp. 43-50, 2015.

[6] M. Deshpande and G. Karypis, "Item-Based Top-n Recommendation Algorithms," *ACM Trans. Information Systems*, vol. 22, no. 1, pp. 143-177, 2004.

[7] D. Dueck and B. Frey, "Probabilistic Sparse Matrix Factorization," *Technical Report PSI TR 2004-023*, Dept. of Computer Science, University of Toronto, 2004.

[8] J. Golbeck, "Computing and Applying Trust in Web-based Social Networks," *PhD Thesis*, University of Maryland College Park, 2005.

[9] P. Gopalan, J. M. Hofman, and D. M. Blei, "Scalable Recommendation with Hierarchical Poisson Factorization," *Proc. 31st Conf. Uncertainty in Artificial Intelligence (UAI'15)*, pp. 326-335, 2015.

[10] M. S. Granovetter, "The Strength of Weak Ties," *The American Journal of Sociology*, vol. 78, no. 6, pp. 1360-1380, 1973.

[11] G. Guo, J. Zhang, and N. Yorke-Smith, "TrustSVD: Collaborative Filtering with Both the Explicit and Implicit Influence of User Trust and of Item Ratings," *Proc. 29th AAAI Conf. Artificial Intelligence (AAAI'15)*, pp. 123-129, 2015.

[12] T. Hofmann, "Latent Semantic Models for Collaborative Filtering," *ACM Trans. Information Systems*, vol. 22, no. 1, pp. 89-115, 2004.

[13] M. Jamali and M. Ester, "TrustWalker: A Random Walk Model for Combining Trust-Based and Item-Based Recommendation," *Proc. 15th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD'09)*, pp. 397-406, 2009.

[14] M. Jamali and M. Ester, "A Matrix Factorization Technique with Trust Propagation for Recommendation in Social Networks," *Proc. 4th ACM Conf. Recommender Systems*, pp. 135-142, 2010.

[15] H. Kautz, B. Selman, and M. Shah, "ReferralWeb: Combining Social Networks and Collaborative Filtering," *Communicatins of the ACM*, vol. 40, no. 3, pp. 63-65, 1997.

[16] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," *Proc. 14th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD'08)*, pp. 426-434, 2008.

[17] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," *Computer*, vol. 42, no. 8, pp. 30-37, 2009.

[18] Y. Koren, "Collaborative Filtering with Temporal Dynamics," *Communications of the ACM*, vol. 53, no. 4, pp. 89-97, 2010.

[19] 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.

[20] N. N. Liu and Q. Yang, "Eigenrank: A Ranking-oriented Approach to Collaborative Filtering," *Proc. 31st Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR'08)*, pp. 83-90, 2008.

[21] H. Ma, T. C. Zhou, M. R. Lyu, and I. King, "Improving Recommender Systems by Incorporating Social Contextual Information," *ACM Trans. Information Systems*, vol. 29, no. 2, pp. 9, 2011.

[22] H. Ma, I. King, and M. R. Lyu, "Learning to Recommend with Explicit and Implicit Social Relations," *ACM Trans. Intelligent Systems and Technology*, vol. 2, no. 3, pp. 29, 2011.

[23] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender Systems with Social Regularization," *Proc. 4th ACM Int'l Conf. Web Search and Data Mining (WSDM'11)*, pp. 287-296, 2011.

[24] P. Massa and P. Avesani, "Trust-aware Collaborative Filtering for Recommender Systems," *Proc. CoopIS, DOA and ODBASE*, pp. 492-508, 2004.

[25] P. Massa and P. Avesani, "Trust-aware Recommender Systems," *Proc. ACM Conf. Recommender Systems*, pp. 17-24, 2007.

[26] J.D.M. Rennie and N. Srebro, "Fast Maximum Margin Matrix Factorization for Collaborative Prediction," *Proc. 22nd Int'l Conf. Machine Learning (ICML'05)*, pp. 713-719, 2005.

[27] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: An Open Architecture for Collaborative Filtering Of Netnews," *Proc. ACM Conf. Computer Supported Cooperative Work*, pp. 175-186, 1994.

[28] P. Resnick, and H. R. Varian, "Recommender Systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56-58, 1997.

[29] R. Salakhutdinov and A. Mnih, "Probabilistic Matrix Factorization," *Proc. 21st Int'l Conf. Neural Information Processing Systems (NIPS'07)*, vol.20, pp. 1257-1264, 2007.

[30] R. Salakhutdinov and A. Mnih, "Bayesian Probabilistic Matrix Factorization Using Markov Chain Monte Carlo," *Proc. 25th Int'l Conf. Machine Learning (ICML'08)*, pp. 880-887, 2008.

[31] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-Based Collaborative Filtering Recommendation Algorithms," *Proc. 10th Int'l Conf. World Wide Web (WWW'01)*, pp. 285-295, 2001.

[32] L. Si and R. Jin, "Flexible Mixture Model for Collaborative Filtering," *Proc. 20th Int'l Conf. Machine Learning (ICML'03)*, pp. 704-711, 2003.

[33] R. Sinha and K. Swearingen, "Comparing Recommendations Made by Online Systems and Friends," *Proc. DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries*, pp. 64-67, 2001.

[34] P. Victor, C. Cornelis, M. De Cock, and P. Pinheiro da Silva, "Gradual Trust and Distrust in Recommender Systems," *Fuzzy Sets and Systems*, vol. 160, no. 10, pp. 1367-1382, 2009.

[35] P. Victor, C. Cornelis, M. De Cock, and A. Teredesai, "Trust- and Distrust-Based Recommendations for Controversial Reviews," *IEEE Intelligent Systems*, vol. 26, no. 1, pp. 48-55, 2011.

[36] J. Wang, A. P. De Vries, and M. J. Reinders, "Unifying User-Based and Item-Based Collaborative Filtering Approaches by Similarity Fusion," *Proc. 29th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR'06)*, pp. 501-508, 2006.

[37] B. Yang, Y. Lei, D. Liu, and J. Liu, "Social Collaborative Filtering by Trust," *Proc. 23rd Int'l Joint Conf. Artificial Intelligence (IJCAI'13)*, pp. 2747-2753, 2013.

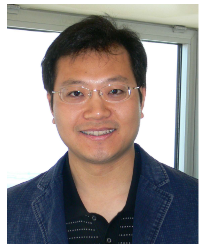
[38] X. Yang, H. Steck, and Y. Liu, "Circle-Based Recommendation in Online Social Networks," *Proc. 18th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD'12)*, pp. 1267-1275, 2012.

[39] Y. Zhang and J. Koren, "Efficient Bayesian Hierarchical User Modeling for Recommendation System," *Proc. 30th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR'07)*, pp. 47-54, 2007.

[40] T. Zhao, C. Li, M. Li, Q. Ding, and L. Li, "Social Recommendation Incorporating Topic Mining and Social Trust Analysis," *Proc. 22nd ACM Int'l Conf. Information and Knowledge Management (CIKM'13)*, pp. 1643-1648, 2013.

[41] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan, "Large-Scale Parallel Collaborative Filtering for the Netflix Prize," *Proc. 4th Int'l Conf. Algorithmic Aspects in Information and Management (AAIM'08)*, pp. 337-348, 2008.

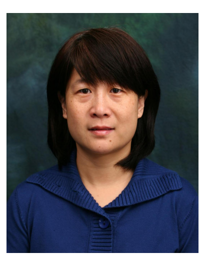
[42] C. N. Ziegler and G. Lausen, "Analyzing Correlation between Trust and User Similarity in Online Communities," *Proc. 2nd Int'l Conf. Trust Management (iTrust'04)*, pp. 251-265, 2004.



Bo Yang is currently a professor in the College of Computer Science and Technology, Jilin University. He received his B.Sc., M.S., and Ph.D. degrees in computer science from Jilin University in 1997, 2000, and 2003, respectively. He is currently the director of Key Laboratory of Symbolic Computation and Knowledge Engineer, Ministry of Education, China. He ever visited several universities and labs as a research fellow or visiting scholar including Computer Science Departments at Hong Kong Baptist University from 2005 to 2006, at Warwick University from 2006 to 2007, at George Mason University from 2014 to 2015, and so on. His current research interests are in the areas of data mining, complex/social network modeling and analysis, multi-agent systems. He is currently working on the topics of discovering structural and dynamical patterns from large-scale and time-evolving networks with applications to Web intelligence, recommendation systems and early detection/control of infectious events. He has published over 70 articles on international journals including IEEE/ACM Transactions, DKE, JAAMAS, KBS, and international conferences including IJCAI, AAAI, ICDM, WI, PAKDD, ASONAM. He has served as the associated editors and peer reviewers for international journals including Web Intelligence and served as the senior PC or PC members for international conferences including IJCAI and AAMAS.

Yu Lei received his M.Eng degree in the School of Computer Science and Technology from Jilin University, Jilin, China. He is currently a Ph.D. student in the Department of Computing, The Hong Kong Polytechnic University, Hum Hong, Hong Kong. His current research interests include recommender systems, collaborative filtering, user behavior modeling, and social media mining.

Jiming Liu (F'11) received the Master of engineering and Ph.D. degrees from McGill University, Montreal, QC, Canada. He is currently the chair professor in Computer Science and the dean of the Faculty of Science at the Hong Kong Baptist University, Kowloon, Hong Kong. He has published many international journal articles and conference papers, and he is the author of research monographs and edited books. He has contributed to the academic community in various capacities, e.g., serving as the Editor-in-Chief of Web Intelligence: An International Journal; an Associate Editor of the IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Systems, Man, and Cybernetics - Part B, Computational Intelligence, etc.; and a member of the editorial board of other international journals. His areas of research are in multi-agent autonomy-oriented computing, Web intelligence, self-organizing/decentralized systems, and real-world complex systems/complex network related problems (e.g., in health and health systems, extreme event management, and sustainability/energy management).



Wenjie Li is currently an associate professor of the Department of Computing at The Hong Kong Polytechnic University. She received the B.Sc. and M.Sc. degrees from Tianjin University, China, and the Ph.D. degree from the Department of Systems Engineering and Engineering Management at the Chinese University of Hong Kong, Hong Kong. Her research interests include natural language processing, text mining, social media analysis, information retrieval, extraction and summarization. She has directed and participated in quite a number of research projects and published over 200 papers in major international journals and conference proceedings, including IEEE TKDE, IEEE TNNLS, ACM TOIS, ACM TALIP, CL, AAAI, ACL, WWW, SIGIR and CIKM. She has served as the information officer of SIGHAN and the associate editor of IJCPOL.