Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

TrustSVD: Collaborative Filtering with Both the Explicit and Implicit Influence of User Trust and of Item Ratings

Guibing Guo, Jie Zhang, Neil Yorke-Smith*
School of Computer Engineering, Nanyang Technological University, Singapore
*American University of Beirut, Lebanon; and University of Cambridge, UK
{gguo1,zhangj}@ntu.edu.sg, *nysmith@aub.edu.lb

Abstract

Collaborative filtering suffers from the problems of data sparsity and cold start, which dramatically degrade recommendation performance. To help resolve these issues, we propose *TrustSVD*, a trust-based matrix factorization technique. By analyzing the social trust data from four real-world data sets, we conclude that not only the explicit but also the implicit influence of both ratings and trust should be taken into consideration in a recommendation model. Hence, we build on top of a state-of-the-art recommendation algorithm SVD++ which inherently involves the explicit and implicit influence of rated items, by further incorporating both the explicit and implicit influence of trusted users on the prediction of items for an active user. To our knowledge, the work reported is the first to extend SVD++ with social trust information. Experimental results on the four data sets demonstrate that our approach TrustSVD achieves better accuracy than other ten counterparts, and can better handle the concerned issues.

Introduction

Incorporating trust into recommender systems has demonstrated potential to improve recommendation performance (Yang et al. 2013; Fang, Bao, and Zhang 2014), and to help mitigate some well-known issues, such as *data sparsity* and *cold start* (Guo, Zhang, and Thalmann 2012). Such trust-aware approaches are developed based on the phenomenon that friends often influence each other by recommending items. However, even the best performance reported by the latest work (Fang, Bao, and Zhang 2014) can be inferior to that of other state-of-the-art models which are merely based on user-item ratings. For instance, a well-performing trust-based model (Yang et al. 2013) obtains 1.0585 on data set Epinions.com in terms of Root Mean Square Error (RMSE), whereas the performance of a user-item baseline (see Koren (2008), Sect. 2.1) can achieve 1.0472 in terms of RMSE.[1]

To investigate this phenomenon, we conduct an empirical trust analysis based on four real-word data sets (FilmTrust, Epinions, Flixster and Ciao) through which two important observations are concluded. First, trust information is also very sparse, yet complementary to rating information. Hence, focusing too much on either one kind of information may achieve only marginal gains in predictive accuracy. Second, users are strongly correlated with their trust neighbors whereas they have a weakly positive correlation with their *trust-alike* neighbors (e.g., friends). Given that very few trust networks exist, it is better to have a more general trust-based model that can well operate on both trust and trust-alike relationships. These observations motivate us to consider both explicit and implicit influence of ratings and of trust in a trust-based model. The influence can be explicit (real values of ratings and trust) or implicit (who rates what (for ratings) and who trusts whom (for trust)). The implicit influence of ratings has been demonstrated useful in providing accurate recommendations (Koren 2008). We will later show that implicit trust can also provide added value over explicit trust.

Thus we propose a novel trust-based recommendation model *TrustSVD*. Our approach builds on top of a state-of-the-art model SVD++ (Koren 2008) where both the explicit and implicit influence of user-item ratings are involved to generate predictions. To the authors' knowledge, our work is the first to extend SVD++ with social trust information. Specifically, on one hand the implicit influence of trust (who trusts whom) can be naturally added to the SVD++ model by extending the user modeling. On the other hand, the explicit influence of trust (trust values) is used to constrain that user-specific vectors should conform to their social trust relationships. This ensures that user-specific vectors can be learned from their trust information even if a few or no ratings are given. In this way, the data sparsity and cold start issues can be better alleviated. Our novel model thus incorporates both explicit and implicit influence of item ratings as well as user trust. In addition, a weighted-λ-regularization technique is used to further avoid over-fitting for model learning. Experimental results on the four real-world data sets demonstrate that our approach achieves significantly better accuracy than other trust-based counterparts as well as other ratings-only well-performing models (ten approaches in total), and is more capable of coping with cold start situations.

Related Work

Trust-aware recommender systems have been widely studied, given that social trust provides an alternative view of user preferences other than item ratings (Guo, Zhang, and

[1] Smaller RMSE values indicate better predictive accuracy. Result reported by the recommendation toolkit MyMediaLite (mymedialite.net/examples/datasets.html).

Yorke-Smith 2014). Specifically, Ma et al. (2008) propose a social regularization method (*SoRec*) by considering the constraint of social relationships. The idea is to share a common user-feature matrix factorized by ratings and by trust. Ma, King, and Lyu (2009) then propose a social trust ensemble method (*RSTE*) to linearly combine a basic matrix factorization model and a trust-based neighborhood model together. Ma et al. (2011) further propose that the active user's user-specific vector should be close to the average of her trusted neighbors, and use it as a regularization to form a new matrix factorization model (*SoReg*). Jamali and Ester (2010) build a new model (*SocialMF*) on top of SoRec by reformulating the contributions of trusted users to the formation of the active user's user-specific vector rather than to the predictions of items. Yang et al. (2013) propose a hybrid method (*TrustMF*) that combines both a truster model and a trustee model from the perspectives of trusters and trustees, that is, both the users who trust the active user and those who are trusted by the user will influence the user's ratings on unknown items. Tang et al. (2013) consider both global and local trust as the contextual information in their model, where the global trust is computed by a separate algorithm. Yao et al. (2014) take into consideration both the explicit and implicit interactions among trusters and trustees in a recommendation model. Fang, Bao, and Zhang (2014) stress the importance of multiple aspects of social trust. They decompose trust into four general factors and then integrate them into a matrix factorization model. All these works have shown that a matrix factorization model regularized by trust outperforms the one without trust. That is, trust is helpful in improving predictive accuracy. However, it is also noted that even the latest work (Fang, Bao, and Zhang 2014) can be inferior to other well-performing ratings-only models. To explain this phenomenon, we next conduct a trust analysis to investigate the value of trust in recommender systems.

Trust Analysis

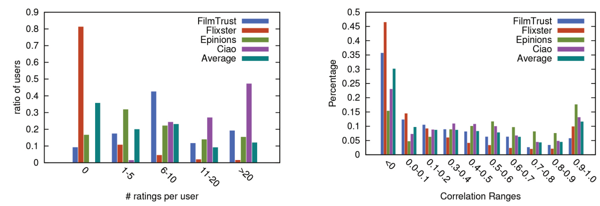
Four data sets are used in our analysis and also our later experiments: Epinions (trustlet.org/wiki/Epinions_datasets), FilmTrust (www.librec.net/datasets.html), Flixster (www.cs.sfu.ca/∼sja25/personal/datasets/) and Ciao (www.public.asu.edu/∼jtang20/datasetcode/truststudy.htm). All the four data sets contain both item ratings and social relationships specified by active users. The ratings in Epinions and Ciao are integers from 1 to 5, while those in the other data sets are real values, i.e., [0.5, 4.0] for FilmTrust, [0.5, 5.0] for Flixster both with step 0.5. Note that trust is asymmetric in Epinions, FilmTrust and Ciao whereas it is symmetric in Flixster. A subset of the Flixster data set is used in this paper to avoid memory-consuming issues. The data set statistics are illustrated in Table 1. Two important observations are concluded from these data sets.

Observation 1 *Trust information is very sparse, yet is complementary to rating information.*

On one hand, as shown in Table 1, the density of trust is much smaller than that of ratings in Epinions, FilmTrust and Flixster whereas trust is only denser than ratings in Ciao. Both ratings and trust are very sparse across all the data sets.

Table 1: Statistics of the four data sets

Feature	Epinions	FilmTrust	Flixster	Ciao
users	~~40,163~~	1,508	53,213	7,375
items	139,738	2,071	18,197	99,746
ratings	664,824	35,497	409,803	280,391
density	0.051%	1.14%	0.04%	0.03%
trusters	33,960	609	47,029	6,792
trustees	49,288	732	47,029	7,297
trusts	487,183	1,853	655,054	111,781
density	0.029%	0.42%	0.03%	0.23%



(a) (b)

Figure 1: (a) The distribution of ratio of users who have issued trust statements w.r.t. the number of ratings that they each have given. (b) The correlations between a user's ratings and those of her trusted neighbors in all the data sets.

In this regard, a trust-aware recommender system that focuses too much on trust (rather than rating) utility is likely to achieve only marginal gains in recommendation performance. In fact, the existing trust-based models consider only the explicit influence of ratings. That is, the utility of ratings is not well exploited. In addition, the sparsity of explicit trust also implies the importance of involving implicit trust in collaborative filtering. On the other hand, trust information is complementary to the rating information. Figure 1a shows that: (1) A portion of users have not rated any items but are socially connected with other users. (2) For the *cold-start* users who have rated few items (less than 5 in our case), trust information can provide a complementary part of source of information with ratio greater than 10% on average. (3) The *warm-start* users who have rated a lot of items (e.g., > 20) are not necessary to specify many other users as trustworthy (12% on the average). Although having differing distributions across the data sets, trust can be a complementary information source to item ratings for recommendations.

This observation motivates us to consider both the explicit and implicit influence of ratings and trust, making better and more use of ratings and trust to resolve the concerned issues.

Observation 2 *A user's ratings have a* ***weakly positive correlation*** *with the average of her social neighbors under the concept of trust-alike relationships, and a* ***strongly positive correlation*** *under the concept of trust relationships.*

Next, we consider the influence of trust in rating prediction, i.e., the influence of trusted neighbors on the active user's rating for a specific item, a.k.a. *social influence*. Specifically, we calculate the Pearson correlation coefficient (PCC) between a user's ratings and the average of her social neighbors. The results are presented in Figure 1b, indicating that: (1) A weakly positive correlation is observed between a user's ratings and the average of the social neighbors in FilmTrust (mean 0.183) and Flixster (0.063). The distributions of the two data sets are similar. Flixster adopts the symmetric friendship relationships whereas trust is directed. Although FilmTrust adopts the concept of trust (with values from 1 to 10), the publicly available data set contains only binary values (such degrading may cause much noise). We regard these relationships as *trust alike*, i.e., the social relationships that are similar with, but weaker (or more noisy) than social trust. (2) Under the concept of trust relationships, on the contrary, a user's ratings are strongly and positively correlated with the average of trusted neighbors. Specifically, a large portion (17.63% in Epinions, 13.14% in Ciao) of user correlations are in the range of $[0.9, 1.0]$, and (resp. 54.70%, 39.14%) of user correlations are greater than 0.5. The average correlation is 0.446 in Epinions, and 0.322 in Ciao. Since PCC values are in the range of $[-1, 1]$, values of 0.446 and 0.322 indicate decent correlations. In the social networks with relatively weak trust-alike relationships (e.g., friendship), implicit influence (i.e., binary relationships) may be more indicative than explicit (but noisy) values for recommendations. Hence, a trust-based model that ignores the implicit influence of ratings and trust may lead to deteriorated performance if being applied to such cases.

The second observation suggests that incorporating both the explicit and implicit influence of ratings and trust may promote the generality of a trust-based model to both trust and trust-alike social relationships. Our approach presented next is constructed based on these two observations.

TrustSVD: A Trust-based Model

Problem Definition

The recommendation problem in this paper is to predict the rating that a user will give to an unknown item, based on both a user-item rating matrix and a user-user trust matrix. Suppose that a recommender system includes m users and n items. Let $R = [r_{u,i}]_{m\times n}$ denote the user-item rating matrix, where each entry $r_{u,i}$ represents the rating given by user u on item i. For clarity, we preserve symbols u, v for users, and i, j for items. Let I_u denote the set of items rated by user u. Let p_u and q_i be a d-dimensional latent feature vector of user u and item i, respectively. The essence of matrix factorization is to find two low-rank matrices: user-feature matrix $P \in \mathbb{R}^{d\times m}$ and item-feature matrix $Q \in \mathbb{R}^{d\times n}$ that can adequately recover the rating matrix R, i.e., $R \approx P^\top Q$, where $P^\top$ is the transpose of matrix P. Hence, the rating on item j for user u can be predicted by the inner product of user-specific vector p_u and item-specific vector q_j, i.e., $\hat{r}_{u,j} = q_j^\top p_u$. In this regard, the main task of recommendations is to predict the rating $\hat{r}_{u,j}$ as close as possible to the ground truth $r_{u,j}$. Formally, we can learn the user- and item-feature matrices by minimizing the following loss (objective) function:

$$\mathcal{L}_r = \frac{1}{2}\sum_u \sum_{j\in I_u} (q_j^\top p_u - r_{u,j})^2 + \frac{\lambda}{2}\Big(\sum_u \|p_u\|_F^2 + \sum_j \|q_j\|_F^2\Big),$$

where $\|\cdot\|_F$ denotes the Frobenius norm, and λ is a parameter to control model complexity and to avoid over-fitting.

On the other hand, suppose that a social network is represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ includes a set of m nodes (users) and $\mathcal{E}$ represents the directed trust relationships among users. We can use the adjacency matrix

social

$T = [t_{u,v}]_{m\times m}$ to describe the structure of edges $\mathcal{E}$, where $t_{u,v}$ indicates the extent to which users u trusts v. We denote p_u and w_v as the d-dimensional latent feature vector of truster u and trustee v, respectively. We limit the trusters in the trust matrix and the active users in the rating matrix to share the same user-feature space in order to bridge them together. Hence, we have truster-feature matrix $P^{d\times m}$ and trustee-feature matrix $W^{d\times m}$. By employing the low-rank matrix approximation, we can recover the trust matrix by $T \approx P^\top W$. Thus, a trust relationship can be predicted by the inner product of a truster-specific vector and a trustee-specific vector $\hat{t}_{u,v} = w_v^\top p_u$. The matrices P and W can be learned by minimizing the following loss function:

$$\mathcal{L}_t = \frac{1}{2}\sum_u \sum_{v\in T_u} (w_v^\top p_u - t_{u,v})^2 + \frac{\lambda}{2}\Big(\sum_u \|p_u\|_F^2 + \sum_v \|w_v\|_F^2\Big),$$

where T_u is the set of users trusted by user u.

The TrustSVD Model

In line with the two observations of the previous section, our TrustSVD model is built on top of a state-of-the-art model known as SVD++ proposed by Koren (2008). The rationale behind SVD++ is to take into consideration user/item biases and the influence of rated items other than user/item-specific vectors on rating prediction. Formally, the rating for user u on item j is predicted by:

$$\hat{r}_{u,j} = b_u + b_j + \mu + q_j^\top \Big(p_u + |I_u|^{-\frac{1}{2}} \sum_{i\in I_u} y_i\Big),$$

均

where b_u, b_j represent the user and item biases, respectively; μ is the global average rating; and y_i denotes the implicit influence of items rated by user u in the past on the ratings of unknown items in the future. Thus, user u's feature vector can be also represented by the set of items she rated, and finally modeled as $\big(p_u + |I_u|^{-\frac{1}{2}} \sum_{i\in I_u} y_i\big)$ rather than simply as p_u. Koren (2008) has shown that integrating implicit influence of ratings can well improve predictive accuracy.

Previously, we have stressed the importance of trust influence for better recommendations, and its potential to be generalized to trust-alike relationships. Hence, we can enhance the trust-unaware SVD++ model by incorporating trust influence. Specifically, the implicit effect of trusted users on item ratings can be considered in the same manner as rated items, given by:

$$\begin{aligned}\hat{r}_{u,j} = {} & b_u + b_j + \mu \\ & + q_j^\top \Big(p_u + |I_u|^{-\frac{1}{2}} \sum_{i\in I_u} y_i + |T_u|^{-\frac{1}{2}} \sum_{v\in T_u} w_v\Big),\end{aligned}$$

where w_v is the user-specific latent feature vector of users (trustees) trusted by user u, and thus $q_j^\top w_v$ can be explained by the ratings predicted by the trusted users, i.e., the influence of trustees on the rating prediction. In other words, the inner product $q_j^\top w_v$ indicates how trusted users influence user u's rating on item j. Similar to ratings, a user's feature vector can be interpreted by the set of users whom she trusts, i.e., $|T_u|^{-\frac{1}{2}} \sum_{v\in T_u} w_v$. Therefore, a user u is further modeled by $\big(p_u + |I_u|^{-\frac{1}{2}} \sum_{i\in I_u} y_i + |T_u|^{-\frac{1}{2}} \sum_{v\in T_u} w_v\big)$ in the social rating networks, considering the influence of both rated items and trusted users. The objective function to minimize is then given as follows:

$$\mathcal{L} = \frac{1}{2}\sum_u \sum_{j\in I_u} \left(\hat{r}_{u,j} - r_{u,j}\right)^2 + \frac{\lambda}{2}\Big(\sum_u b_u^2 + \sum_j b_j^2 + \sum_u \|p_u\|_F^2 + \sum_j \|q_j\|_F^2 + \sum_i \|y_i\|_F^2 + \sum_v \|w_v\|_F^2\Big).$$

To reduce the model complexity, we use the same regularization parameter λ for all the variables. Finer control and tuning can be achieved by assigning separate regularization parameters to different variables, but it may result in great complexity when comparing with different models.

In addition, as explained earlier, we constrain that the user-specific vectors decomposed from the rating matrix and those decomposed from the trust matrix share the same feature space in order to bridge both matrices together. In this way, these two types of information can be exploited in a unified recommendation model. Specifically, we can regularize the user-specific vectors p_u by recovering the social relationships with other users. The new objective function (without the other regularization terms) is given by:

$$\mathcal{L} = \frac{1}{2}\sum_u \sum_{j\in I_u} \left(\hat{r}_{u,j} - r_{u,j}\right)^2 + \frac{\lambda_t}{2}\sum_u \sum_{v\in T_u} \left(\hat{t}_{u,v} - t_{u,v}\right)^2,$$

where $\hat{t}_{u,v} = w_v^\top p_u$ is the predicted trust between users u and v, and λ_t controls the degree of trust regularization.

Further, as suggested by Yang et al. (2013), a technique called weighted-λ-regularization can be used to help avoid over-fitting when learning parameters. In particular, they consider more penalties for the users who rated more items and for the items which received more ratings. However, we argue that such consideration may force the model to be more biased towards popular users and items. Instead, in this paper we adopt a distinct strategy that the popular users and items should be less penalized (due to smaller chance to be over-fitted), and cold-start users and *niche items* (those receiving few ratings) should be more regularized (due to greater chance to be over-fitted). Therefore, the new loss function to minimize is obtained as follows:

$$\begin{aligned}\mathcal{L} = &\frac{1}{2}\sum_u \sum_{j\in I_u} \left(\hat{r}_{u,j} - r_{u,j}\right)^2 + \frac{\lambda_t}{2}\sum_u \sum_{v\in T_u} \left(\hat{t}_{u,v} - t_{u,v}\right)^2 \\ &+ \frac{\lambda}{2}\sum_u |I_u|^{-\frac{1}{2}} b_u^2 + \frac{\lambda}{2}\sum_j |U_j|^{-\frac{1}{2}} b_j^2 \\ &+ \sum_u \left(\frac{\lambda}{2}|I_u|^{-\frac{1}{2}} + \frac{\lambda_t}{2}|T_u|^{-\frac{1}{2}}\right)\|p_u\|_F^2 \\ &+ \frac{\lambda}{2}\sum_j |U_j|^{-\frac{1}{2}}\|q_j\|_F^2 + \frac{\lambda}{2}\sum_i |U_i|^{-\frac{1}{2}}\|y_i\|_F^2 \\ &+ \frac{\lambda}{2}|T_v^+|^{-\frac{1}{2}}\|w_v\|_F^2,\end{aligned} \tag{1}$$

where U_j, U_i are the set of users who rate items j and i, respectively; and T_v^+ is the set of users who trust user v. Since the active user has rated a number of items and specified other users as trustworthy, the penalization on user-specific vector p_u takes into account both cases.

Model Learning

To obtain a local minimization of the objective function given by Equation 1, we perform the following gradient descents on b_u, b_j, p_u, q_j, y_i and w_v for all the users and items.

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial b_u} &= \sum_{j\in I_u} e_{u,j} + \lambda|I_u|^{-\frac{1}{2}} b_u \\ \frac{\partial\mathcal{L}}{\partial b_j} &= \sum_{u\in U_j} e_{u,j} + \lambda|U_j|^{-\frac{1}{2}} b_j \\ \frac{\partial\mathcal{L}}{\partial p_u} &= \sum_{j\in I_u} e_{u,j} q_j + \lambda_t \sum_{v\in T_u} e_{u,v} w_v \\ &\quad + \left(\lambda|I_u|^{-\frac{1}{2}} + \lambda_t|T_u|^{-\frac{1}{2}}\right)p_u \\ \frac{\partial\mathcal{L}}{\partial q_j} &= \sum_{u\in U_j} e_{u,j}\Big(p_u + |I_u|^{-\frac{1}{2}}\sum_{i\in I_u} y_i \\ &\quad + |T_u|^{-\frac{1}{2}}\sum_{v\in T_u} w_v\Big) + \lambda|U_j|^{-\frac{1}{2}} q_j \\ \forall i\in I_u, \frac{\partial\mathcal{L}}{\partial y_i} &= \sum_{j\in I_u} e_{u,j}|I_u|^{-\frac{1}{2}} q_j + \lambda|U_i|^{-\frac{1}{2}} y_i \\ \forall v\in T_u, \frac{\partial\mathcal{L}}{\partial w_v} &= \sum_{j\in I_u} e_{u,j}|T_u|^{-\frac{1}{2}} q_j + \lambda_t e_{u,v} p_u + \lambda|T_v^+|^{-\frac{1}{2}} w_v\end{aligned} \tag{2}$$

where $e_{u,j} = \hat{r}_{u,j} - r_{u,j}$ indicates the rating prediction error for user u on item j, and $e_{u,v} = \hat{t}_{u,v} - t_{u,v}$ is the trust prediction error for user u towards user v.

In the cold-start situations where users may have only rated a few items, the decomposition of trust matrix can help to learn more reliable user-specific latent feature vectors than ratings-only matrix factorization. In the extreme case where there are no ratings at all for some users, Equation 2 ensures that the user-specific vector can be trained and learned from the trust matrix. In this regard, incorporating trust in a matrix factorization model can alleviate the cold start problem. By considering both explicit and implicit influence of trust rather than either one alone, our model can better utilize trust to further mitigate the concerned issues.

Complexity Analysis

The computational time of learning the TrustSVD model is mainly taken by evaluating the objective function $\mathcal{L}$ and its gradients against feature vectors (variables). The time to compute the objective function $\mathcal{L}$ is $O(d|R| + d|T|)$, where d is the dimensionality of the feature space, and $|R|, |T|$ refer to the number of observed entries. Due to the sparsity of rating and trust matrices, the values will be much smaller than the matrix cardinality. The computational complexities for gradients $\frac{\partial\mathcal{L}}{\partial b_u}, \frac{\partial\mathcal{L}}{\partial b_j}, \frac{\partial\mathcal{L}}{\partial p_u}, \frac{\partial\mathcal{L}}{\partial q_j}, \frac{\partial\mathcal{L}}{\partial y_i}, \frac{\partial\mathcal{L}}{\partial w_v}$ in Equation 2 are $O(d|R|), O(d|R|), O(d|R| + d|T|), O(d|R| + d|T|), O(d|R|k)$ and $O(d|R|p + d|T|p)$, where k, p are the average number of ratings and trust statements received by an item and a user, respectively. Hence, the overall computational complexity in one iteration is $O(d|R|c + d|T|c)$, where $c = \max(p, k)$. Due to $c \ll |R|$ or $|T|$, the overall computational time is linear with respect to the number of observations in the rating and trust matrices. To sum up, our model has potential to scale up to large-scale data sets.

Experiments and Results

Data Sets. The four data sets presented in Table 1 are used.

Cross-validation. We use 5-fold cross-validation for learning and testing. Specifically, we randomly split each data set into five folds and in each iteration four folds are used as the

Table 2: Performance comparison in two testing views, where * indicates the best performance among all the other methods, and column 'Improve' indicates the relative improvements that our approach TrustSVD achieves relative to the * results.

All	**Metrics**	**UserAvg**	**ItemAvg**	**PMF**	**RSTE**	**SoRec**	**SoReg**	**SocialMF**	**TrustMF**	**SVD++**	**TrustSVD**	**Improve**
FilmTrust	MAE	0.636	0.725	0.714	0.628	0.628	0.674	0.638	0.631	0.613*	**0.609**	0.65%
$d = 5$	RMSE	0.823	0.927	0.949	0.810	0.810	0.878	0.837	0.810	0.804*	**0.789**	1.87%
	MAE	0.636	0.725	0.735	0.640	0.638	0.668	0.642	0.631	0.611*	**0.607**	0.65%
$d = 10$	RMSE	0.823	0.927	0.968	0.835	0.831	0.875	0.844	0.819	0.802*	**0.787**	1.87%
Epinions	MAE	0.930	0.928	0.979	0.950	0.882	0.994	0.825	0.818	0.818*	**0.804**	1.71%
$d = 5$	RMSE	1.203	1.094	1.290	1.196	1.114	1.315	1.070	1.069	1.057*	**1.043**	1.32%
	MAE	0.930	0.928	0.909	0.958	0.884	0.932	0.826	0.819	0.818*	**0.805**	1.59%
$d = 10$	RMSE	1.203	1.094	1.197	1.278	1.142	1.232	1.082	1.095	1.057*	**1.044**	1.23%
Flixster	MAE	0.729*	0.858	0.814	0.751	0.750	0.820	0.770	0.890	0.794	**0.726**	0.41%
$d = 5$	RMSE	0.979*	1.088	1.076	0.975	0.974	1.087	0.994	1.146	1.062	**0.948**	3.17%
	MAE	0.729*	0.858	0.769	0.784	0.785	0.785	0.784	1.116	0.821	**0.727**	0.27%
$d = 10$	RMSE	0.979*	1.088	1.009	1.015	1.018	1.034	1.009	1.441	1.091	**0.950**	2.97%
Ciao	MAE	0.781	0.760	0.920	0.767	0.765	0.899	0.749	0.742*	0.752	**0.723**	2.56%
$d = 5$	RMSE	1.031	1.026	1.206	1.020	1.013	1.183	0.981*	0.983	1.013	**0.955**	2.65%
	MAE	0.781	0.760	0.822	0.763	0.761	0.815	0.749*	0.753	0.748	**0.723**	3.47%
$d = 10$	RMSE	1.031	1.026	1.078	1.013	1.010	1.076	0.976*	1.014	1.001	**0.956**	2.05%

Cold Start	**Metrics**	**UserAvg**	**ItemAvg**	**PMF**	**RSTE**	**SoRec**	**SoReg**	**SocialMF**	**TrustMF**	**SVD++**	**TrustSVD**	**Improve**
FilmTrust	MAE	0.709	0.722	0.814	0.680	0.670*	0.881	0.697	0.674	0.677	**0.661**	1.34%
$d = 5$	RMSE	0.979	0.911	1.079	0.884	0.857*	1.104	0.916	0.867	0.897	**0.853**	0.47%
	MAE	0.709	0.722	0.767	0.674	0.668*	0.771	0.680	0.687	0.680	**0.663**	0.75%
$d = 10$	RMSE	0.979	0.911	1.009	0.900	0.897*	1.034	0.907	0.900	0.905	**0.853**	4.91%
Epinions	MAE	1.047	0.852*	1.451	1.051	0.892	1.398	0.884	0.891	0.889	**0.868**	-1.88%
$d = 5$	RMSE	1.430	1.127	1.770	1.266	1.138	1.735	1.133	1.125*	1.162	**1.105**	1.78%
	MAE	1.047	0.852	1.153	0.981	0.846*	1.139	0.857	0.853	0.891	**0.868**	-2.60%
$d = 10$	RMSE	1.430	1.127*	1.432	1.313	1.180	1.437	1.152	1.176	1.166	**1.108**	1.69%
Flixster	MAE	0.869	0.906	1.097	0.872	0.872	1.058	0.881	0.901	0.868*	**0.855**	1.50%
$d = 5$	RMSE	1.155	1.114	1.390	1.097	1.096*	1.358	1.103	1.138	1.122	**1.066**	2.74%
	MAE	0.869*	0.906	0.949	0.889	0.892	0.951	0.884	0.976	0.869*	**0.858**	1.27%
$d = 10$	RMSE	1.155	1.114	1.206	1.137	1.144	1.218	1.112*	1.328	1.112*	**1.070**	3.78%
Ciao	MAE	0.829	0.735*	1.033	0.957	0.789	1.173	0.774	0.752	0.759	**0.729**	0.82%
$d = 5$	RMSE	1.138	1.005	1.334	1.113	0.998	1.430	1.001	0.954*	1.039	**0.953**	0.10%
	MAE	0.829	0.735	0.926	0.803	0.730*	0.949	0.741	0.770	0.749	**0.721**	1.23%
$d = 10$	RMSE	1.138	1.005	1.191	1.014	1.031	1.214	0.978*	1.096	1.020	**0.962**	1.64%

training set and the remaining fold as the test set. Five iterations will be conducted to ensure that all folds are tested. The average test performance is given as the final result.

Evaluation Metrics. We adopt two well-known metrics to evaluate predictive accuracy, namely mean absolute error (MAE) and root mean square error (RMSE), defined by:

$$\text{MAE} = \frac{\sum_{u,j} |\hat{r}_{u,j} - r_{u,j}|}{N}, \ \text{RMSE} = \sqrt{\frac{\sum_{u,j} (\hat{r}_{u,j} - r_{u,j})^2}{N}}$$

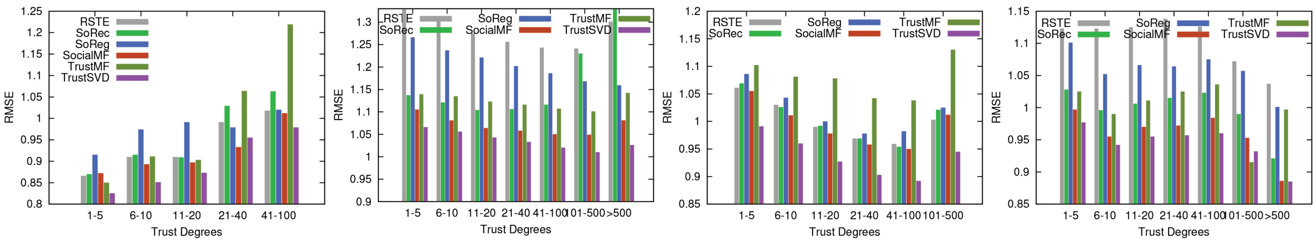
where N is the number of test ratings. Smaller values of MAE and RMSE indicate better predictive accuracy.

Testing Views. Two data set views are created for testing. First, the *All* view indicates that all ratings are used as the test set. Second, the *Cold Start* view means that only the users who rate less than five items will be involved in the test set. Similar testing views are also defined and used in (Guo, Zhang, and Thalmann 2012; Yang et al. 2013).

Comparison Methods. Three kinds of approaches are compared with our method **TrustSVD**[2]: (1) Baselines: **UserAvg** and **ItemAvg** make predictions by the average of ratings that are given by the active user and received by the target item, respectively; (2) Trust-based models: **SoRec** (Ma et al. 2008), **RSTE** (Ma, King, and Lyu 2009), **SoReg** (Ma et al. 2011), **SocialMF** (Jamali and Ester 2010), **TrustMF** (Yang et al. 2013) and **Fang's** (Fang, Bao, and Zhang 2014); (3) Ratings-only state-of-the-art models: **PMF** (Salakhutdinov and Mnih 2008) and **SVD++** (Koren 2008).

Parameter Settings. The optimal experimental settings for each method are determined either by our experiments or suggested by previous works. Specifically, the common settings are $\lambda = 0.001$, and the number of latent features $d = 5/10$, the same as all the previous trust-based models. The other settings are: (1) RSTE: $\alpha = 0.4$ for Epinions, and 1.0 for the others; (2) SoRec: $\lambda_c = 0.1, 1.0, 0.001, 0.01$ corresponding to FilmTrust, Epinions, Flixster and Ciao, respectively; (3) SoReg: $\beta = 1.0$ for Flixster and 0.1 for the others; (4) SocialMF, TrustMF: $\lambda_t = 1$; (5) SVD++: $\lambda = 0.1, 0.35, 0.03, 0.1$ (resp.); (6) TrustSVD: $\lambda = 1.2, \lambda_t = 0.9$ for FilmTrust, $\lambda = 0.9, \lambda_t = 0.5$ for Epinions, $\lambda = 0.8, \lambda_t = 0.5$ for Flixster, and $\lambda = 0.5, \lambda_t = 1$ for Ciao.

[2] Source code is included in the LibRec library at www.librec.net.



(a) FilmTrust (b) Epinions (c) Flixster (d) Ciao

Figure 2: Performance comparison on users with different trust degrees ($d = 10$) [best viewed in color]

Table 3: Comparing with (Fang, Bao, and Zhang 2014)

Fang's vs. **TrustSVD**	**Epinions** $d=5$	**Epinions** $d=10$	**Ciao** $d=5$	**Ciao** $d=10$	**FilmTrust** $d=5$	**FilmTrust** $d=10$
MAE	0.806 **0.804**	0.814 **0.805**	0.737 **0.723**	0.745 **0.723**	0.616 **0.609**	0.625 **0.607**
RMSE	1.047 **1.043**	1.059 **1.044**	0.972 **0.955**	0.985 **0.956**	0.793 **0.789**	0.810 **0.787**

Comparison with other models. The experimental results are presented in Table 2. For all the comparison methods in the testing view of *All*, SVD++ outperforms the other comparison methods in FilmTrust and Epinions, and UserAvg performs the best in Flixster. This implies that these trust-based approaches cannot always beat other well-performing ratings-only approaches, and even simple baselines in trust-alike networks (i.e., FilmTrust and Flixster). Only in Ciao, trust-based approach (SocialMF) gives the best performance. On the contrary, our approach TrustSVD is consistently superior to the best approach among the others across all the data sets. Although the percentage of relative improvements are small, Koren (2010) has pointed out that even small improvements in MAE and RMSE may lead to significant differences of recommendations in practice.

For all the comparison methods in the testing view of *Cold Start*, SoRec and SVD++ perform respectively the best in FilmTrust and Flixster (trust-alike), while no single approach works the best in Epinions and Ciao (trust). Generally, our approach performs better than the others both in trust and trust-alike relationships. Although some exceptions are observed in Epinions in terms of MAE, TrustSVD is more powerful in terms of RMSE. Since all the trust-based models aim to optimize the square errors between predictions and real values, RMSE is more indicative than MAE, and thus TrustSVD still has best performance overall.

Besides the above-compared approaches, some new trust-based models have been proposed recently. The most relevant model is presented in (Fang, Bao, and Zhang 2014); for clarity, we denote it by *Fang's*. It is reported to perform better than other trust-based models and than SVD++ (except in Ciao). Table 3 shows the comparison between Fang's and our approach TrustSVD. The results of Fang's approach are reported in (Fang, Bao, and Zhang 2014), and directly re-used in our paper. Note that we sampled more data for Flixster than Fang's, and thus their experimental results are not comparable. Table 3 clearly shows that our approach performs better than Fang's in terms of both MAE and RMSE.

One more observation from Tables 2 and 3 is that the performance of TrustSVD when $d = 5$ is very close to that when $d = 10$, indicating the reliability of our approach with respect to the feature dimensionality. We ascribe this feature to the consideration of both the explicit and implicit influence of ratings and trust in a unified recommendation model.

In conclusion, the experimental results indicate that our approach TrustSVD outperforms the other methods in predicting more accurate ratings, and that its performance is reliable with different number of latent features.

Comparison in trust degrees. Another series of experiments are conducted to investigate the performance on users with different trust degrees. The trust degrees refer to the number of trusted neighbors specified by a user. We split the trust degrees into seven categories: 1-5, 6-10, 11-20, 21-40, 41-100, 101-500, >500. The results of trust-based models are illustrated in Figure 2, where we only present the performance in RMSE when $d = 10$ due to space limitation; similar trends in other cases. In general, our approach TrustSVD consistently achieves the best performance in the cases of different trust degrees. Paired t-tests (confidence 0.95) shows that such improvements are statistically significant ($p<10^{-5}$) in three data sets but not in FilmTrust (details are omitted to save space). Two possible explanations can be made: (1) FilmTrust has a smaller data size than the others; (2) due to the noise arising from converting real-valued trust to binary trust, the trust may be less useful in FilmTrust than in the others as indicated in (Fang, Bao, and Zhang 2014).

Conclusion and Future Work

This paper proposed a novel trust-based matrix factorization model which incorporated both rating and trust information. Our analysis of trust in four real-world data sets indicated that trust and ratings are complementary to each other, and both pivotal for more accurate recommendations. Our novel approach, TrustSVD, takes into account both the explicit and implicit influence of ratings and trust information when predicting ratings of unknown items. A weighted-λ-regularization technique was adapted and used to further regularize the user- and item-specific latent feature vectors. Comprehensive experimental results showed that our approach TrustSVD outperformed both trust- and ratings-based methods in predictive accuracy across different testing views and across users with different trust degrees. For future work, we intend to further improve the proposed model by considering both the influence of trusters and trustees.

Acknowledgements

Guibing Guo thanks the Institute for Media Innovation for a PhD grant at Nanyang Technological University, Singapore.

The work is also partially supported by the MoE AcRF Tier 2 Grant M4020110.020.

References

Fang, H.; Bao, Y.; and Zhang, J. 2014. Leveraging decomposed trust in probabilistic matrix factorization for effective recommendation. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*, 30–36.

Guo, G.; Zhang, J.; and Thalmann, D. 2012. A simple but effective method to incorporate trusted neighbors in recommender systems. In *Proceedings of the 20th International Conference on User Modeling, Adaptation and Personalization (UMAP)*, 114–125.

Guo, G.; Zhang, J.; and Yorke-Smith, N. 2014. Leveraging multiviews of trust and similarity to enhance clustering-based recommender systems. *Knowledge-Based Systems (KBS)* in press.

Jamali, M., and Ester, M. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the 4th ACM Conference on Recommender Systems (RecSys)*, 135–142.

Koren, Y. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 426–434.

Koren, Y. 2010. Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 4(1):1:1–1:24.

Ma, H.; Yang, H.; Lyu, M.; and King, I. 2008. SoRec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 931–940.

Ma, H.; Zhou, D.; Liu, C.; Lyu, M. R.; and King, I. 2011. Recommender systems with social regularization. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM)*, 287–296.

Ma, H.; King, I.; and Lyu, M. 2009. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 203–210.

Salakhutdinov, R., and Mnih, A. 2008. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*, volume 20, 1257–1264.

Tang, J.; Hu, X.; Gao, H.; and Liu, H. 2013. Exploiting local and global social context for recommendation. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, 2712–2718.

Yang, B.; Lei, Y.; Liu, D.; and Liu, J. 2013. Social collaborative filtering by trust. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, 2747–2753.

Yao, W.; He, J.; Huang, G.; and Zhang, Y. 2014. Modeling dual role preferences for trust-aware recommendation. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR)*, 975–978.