

# Power-law Distribution Aware Trust Prediction

Xiao Wang<sup>1,2,\*</sup>, Ziwei Zhang<sup>2,\*</sup>, Jing Wang<sup>3,†</sup>, Peng Cui<sup>2</sup>, Shiqiang Yang<sup>2</sup>

<sup>1</sup>School of Computer Science, Beijing University of Posts and Telecommunications, China

<sup>2</sup>Department of Computer Science and Technology, Tsinghua University, China

<sup>3</sup>Graduate School of Information Science and Technology, The University of Tokyo, Japan.  
wangxiao\_cv@tju.edu.cn, zw-zhang16@mails.tsinghua.edu.cn, jing\_wang@mist.i.u-tokyo.ac.jp,  
cuip@tsinghua.edu.cn, yangshq@tsinghua.edu.cn

## Abstract

Trust prediction, aiming to predict the trust relations between users in a social network, is a key to helping users discover the reliable information. Many trust prediction methods are proposed based on the low-rank assumption of a trust network. However, one typical property of the trust network is that the trust relations follow the power-law distribution, i.e., few users are trusted by many other users, while most tail users have few trustors. Due to these tail users, the fundamental low-rank assumption made by existing methods is seriously violated and becomes unrealistic. In this paper, we propose a simple yet effective method to address the problem of the violated low-rank assumption. Instead of discovering the low-rank component of the trust network alone, we learn a sparse component of the trust network to describe the tail users simultaneously. With both of the learned low-rank and sparse components, the trust relations in the whole network can be better captured. Moreover, the transitive closure structure of the trust relations is also integrated into our model. We then derive an effective iterative algorithm to infer the parameters of our model, along with the proof of correctness. Extensive experimental results on real-world trust networks demonstrate the superior performance of our proposed method over the state-of-the-arts.

## 1 Introduction

Nowadays, the growth of social media enables online users to share their opinions and information in a more convenient way. Trust prediction [Tang and Liu, 2015], which aims to predict the potential trust relations between users given the observed trust relations, provides an effective solution for a user to discover the relevant information from so much user-generated content in social networks. For example, a user can directly accept information from his/her trustees, so as to

\*These authors contributed equally to this work.

†The corresponding author.

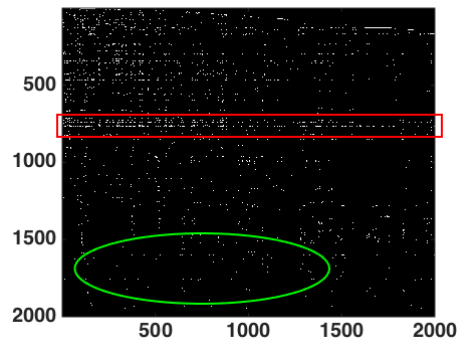


Figure 1: Illustration of the trust network “Ciao”. White dot means there is a trust relation. Obviously, the relations in the red boxes (dense part) show a low-rank structure, while the ones in the green circle show a sparse structure.

avoid spending plenty of time on collecting reliable information. Benefited from this, various trust related applications, such as trust-aware recommender system [Tang *et al.*, 2012] and trust based visualization [O’Donovan *et al.*, 2007], can also be further improved.

The trust network structure is one of the most important available sources for trust prediction, which has been broadly exploited by most existing methods. In a trust network, if user A (trustor) trusts user B (trustee), then there is a trust link (weighted/unweighted) from user A to user B; otherwise, there is no link between them. So in order to well predict the potential trust relations, the inherent properties of a trust network and its topology information should be promoted more attention, for example, the transitivity of a trust network [Guha *et al.*, 2004] and the neighborhood structure [Xiong and Liu, 2004; Zhou and Hwang, 2007]. Recently, many low-rank based models, aiming to learn the low-rank structure of the trust network, are proposed and achieve promising results, e.g., the matrix factorization based model [Tang *et al.*, 2013; Zheng *et al.*, 2014] and rank-k matrix recovery based model [Huang *et al.*, 2013].

One typical property of the real world trust network is the power-law distribution, i.e., some head users have a large

number of trust links with other users, while many tail users only have few trust links with others. This property indicates that a real trust network actually consists of two parts, i.e., the dense part mainly formed by trust relations of head users, and the sparse part mainly formed by the relations of tail users. For a clearer view, here we extracted a subnetwork of the Ciao network [Tang and Liu, 2015] as an example, shown in Figure.1. The rows with red color are the head user structures and the rows with yellow color are the tail user structures. Usually there are few head users and they are with many links, so their structures show low rank property, while for many tail users with few links, their structures show sparsity property. It is well known that the sparse part usually increases the rank of a matrix [Xu *et al.*, 2016], which implies that the fundamental low-rank assumption of the network structure is violated seriously. We further provide a formal approach to establish it. We use SVD with different ranks to reconstruct the trust matrix (Advogato [Massa *et al.*, 2009], Ciao [Tang and Liu, 2015], and Epinions [Tang and Liu, 2015]) and check the relation between the reconstruction error and the rank. Moreover, for better comparison, we generate a  $1000 \times 1000$  matrix with rank 50 and test the errors. As in Figure 2, the horizontal axis indicates the rank, and the vertical axis indicates the normalized reconstruction error. We find that with the rank decreasing, the normalized errors in (b)(c)(d) decrease smoothly, and no obvious inflection points are in the curves. However, the curve in (a) shows a different trend, i.e., after that rank, the error becomes zero and does not change anymore. This indicates that the trust matrix shows different behavior with low-rank matrix, and it should be high-rank. Previous low-rank based models mainly capture the inherent structure, i.e., the dense part, while, the relations of tail users are usually disregarded as the useless or redundant information. So the results are probably biased towards these head users, and the performance of the tail users is hindered. The fact that these large proportions of tail users cannot be well approximated by any linear low dimensional basis is still largely ignored by previous trust prediction literatures. Finally, neglect or improper formulation of the tail users can result in approximating the network structure inaccurately and thereby affecting the quality of a trust prediction method.

Moreover, the observed trust relations are usually extremely rare. For example, the sparsity of Advogato, Ciao, and Epinions, i.e., the ratio of the observed trust relations to all the possible relations, is 0.1195%, 0.2055%, and 0.4135%, respectively. It is challenging to predict the trust relations well with so limited observed links. An alternative solution is to consider the high-order structure [Wang *et al.*, 2017b]. Taking the co-citation trust structure as an example [Guha *et al.*, 2004], it demonstrates that if users A and B trust some common neighbors, then user A is more likely to trust another user that user B trusts. The incorporation of the high-order structure can provide effective and rich information to alleviate the sparsity issue. Therefore, combining the low-rank based model with the high order structure holds a great potential for trust prediction.

In this paper, we propose a matrix factorization based trust prediction method which deals with two typical properties of

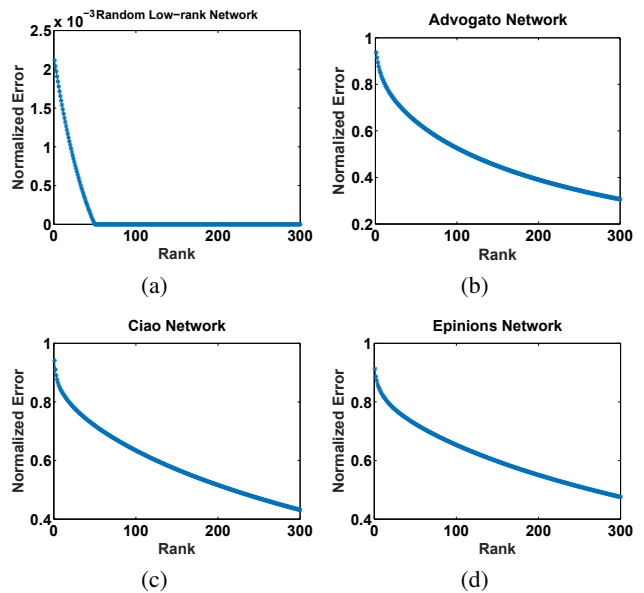


Figure 2: The comparison of low-rank and high-rank properties of trust networks.

the trust network, i.e., the power-law distribution and the network sparsity. Different from the existing methods which only learn a low-rank structure to approximate the original network, we add another sparse term to specifically model the tail users. With both of the low-rank and the sparse terms, the trust relations in a whole network can be well captured. Meanwhile, the transitive closure structure of trust, a widely used high-order structure, is incorporated into the trust matrix. An effective iterative algorithm is also derived to infer the parameters of our model, and its convergence property is analyzed. To summarize, we make the following contributions:

- We studied an important problem in trust prediction, i.e., the violated low-rank structure in a trust network, which makes the traditional low-rank based model become practical and further improves the performance.
- We proposed an effective model which is able to capture the low-rank and the sparse structures of a trust network simultaneously, and theoretically analyzed the correctness of the proposed optimization algorithm.
- We conducted extensive evaluations for our proposed model, as well as the parameter analysis, which demonstrates its effectiveness.

## 2 Related Work

Trust, also related with reputation system, has been widely investigated in the past years. An elaborate review and more discussions can be found in [Josang and Ismail, 2002; Sherchan *et al.*, 2013; Hendriks *et al.*, 2015; Tang and Liu, 2015]. Here we mainly follow the concepts and settings in [Tang and Liu, 2015], and discuss the most related works, i.e., the low-rank based trust prediction models.

Matrix factorization is one of the most widely employed low-rank models. The basic idea is to factorize the trust matrix into the low-rank trustor-specific and trustee-specific matrices, and then various prior knowledge or additional user-generated content can be incorporated. Specifically, [Tang *et al.*, 2013] studies the homophily effect in trust relations, and proposes the homophily regularized matrix factorization model. Then [Yao *et al.*, 2014] explores the transitivity, multi-aspect and trust bias properties of trust network in matrix factorization. Further, [Zheng *et al.*, 2014] considers both the transitivity and similarity factors, especially they introduced the trust rating distribution similarity in the model. Social status theory suggests that users with lower social statuses are more likely to trust users with high statuses, which is considered by [Wang *et al.*, 2015]. Inspired by psychology and sociology, the emotional information is incorporated by [Beigi *et al.*, 2016]. By introducing the user-user positive and negative emotion matrices, they modeled the findings that users with positive/negative emotions are more likely to have trust/distrust relations.

Another low-rank based model follows the idea of matrix completion. For instance, [Huang *et al.*, 2013] proposes a robust rank-k matrix completion method, which explicitly seeks a matrix with the exact rank, so that the low-rank of the matrix recovered can be guaranteed. Instead of using trace norm, [Wang *et al.*, 2017a] propose a max-norm, which is a tighter approximation to the rank function, to learn the low-rank structure of the trust matrix.

However, for all the aforementioned low-rank based methods, due to the presence of a large amount of tail users in the real world trust networks, their fundamental low rank assumption is seriously violated, leading to large approximation error.

### 3 The Proposed Model

Given  $n$  users, we use a trust matrix  $\mathbf{A} = [a_{ij}] \in \mathbb{R}^{n \times n}$  to denote the trust relations between them, where  $a_{ij} = 1$  if we observe that user  $i$  trusts user  $j$ , otherwise  $a_{ij} = 0$ . One reasonable approach to interpret  $\mathbf{A}$  is based on its low-rank assumption, i.e., it can be well approximated in a low-dimensional subspace, which has been well justified by many previous literatures. A very popular model is based on the matrix factorization:

$$\min_{\mathbf{U}, \mathbf{V}} \|\mathbf{A} - \mathbf{UVU}^T\|_F^2 + \lambda_1 \|\mathbf{U}\|_F^2 + \lambda_2 \|\mathbf{V}\|_F^2, \quad (1)$$

where  $\mathbf{U} \in \mathbb{R}^{n \times k}$  is the latent  $k$ -dimensional representations of users, and  $\mathbf{V} \in \mathbb{R}^{k \times k}$  captures the compact correlations among  $\mathbf{U}$ , and  $\lambda_1$  and  $\lambda_2$  are positive parameters for adjusting the contribution of corresponding terms. This model has been widely employed and it is flexible to be combined with other domain knowledge, such as homophily [Tang *et al.*, 2013] and social status theory [Wang *et al.*, 2015].

#### 3.1 Modeling the Tail Users

In reality, the presence of tail users in  $\mathbf{A}$  actually violates the low-rank assumption, so that the learned predictor  $\mathbf{UVU}^T$  may be far from the true  $\mathbf{A}$ . Under such circumstance, the

performance of all the methods based on model (1) is seriously influenced. As mentioned before, the trust network actually consists of the dense and sparse parts. Based on this fact, we do not uniformly model all the users in a low-dimensional space. Instead, we deal with head users and tail users separately. We decompose the trust matrix  $\mathbf{A}$  to

$$\mathbf{A} = \mathbf{UVU}^T + \mathbf{S}, \quad (2)$$

where  $\mathbf{S} = [s_{ij}] \in \mathbb{R}^{n \times n}$  is a sparse matrix capturing the correlations between tail users. Usually,  $L_1$  norm regularization, i.e.,  $\|\mathbf{S}\|_1 = \sum_{i,j} |s_{ij}|$ , is used to encourage the sparsity. Please note that because it is very tricky and difficult to accurately define the head and tail users from all the users, i.e., the boundary between head and tail users is difficult to set, we do not explicitly distinguish them, so we do not specifically model the relations between them. Alternatively, our model is designed in terms of the structures of head and tail users (not the relations), respectively, that is to say, “ $\mathbf{UVU}^T$ ” is mainly to capture the head user structures, and “ $\mathbf{S}$ ” is mainly to capture the tail user structures. But as suggested in [Zou and Hastie, 2005], the elastic net regularization, which adds another smooth term such that both of the sparsity and a group effect are encouraged, often outperforms the lasso. Thus, we employ the elastic net regularization term for  $\mathbf{S}$ :

$$\|\mathbf{S}\|_F^2 + \eta \|\mathbf{S}\|_1, \quad (3)$$

where  $\eta > 0$  is to balance the contributions of the two terms. As can be seen, the elastic net penalty is a convex combination of the lasso and ridge penalty, and thus has their characteristics.

#### 3.2 Modeling Transitive Closure Structure

Directly factorizing  $\mathbf{A}$  just provides little available information. For two users with no link, it does not imply these two users have no trust relation. So it is oversimplified to factorize  $\mathbf{A}$  by taking the observed trust links into account alone. Transitive closure structure is desired for further improving the performance. The transitivity property suggests that if user  $A$  trusts user  $B$ , and user  $B$  trusts user  $C$ , then  $A$  is probably to trust  $C$ , so based on this, we use  $\mathbf{A}^2$  to represent the high order structure. The final trust matrix can be obtained by  $\mathbf{A} \leftarrow \mathbf{A} + \beta \mathbf{A}^2$ , where  $\beta > 0$  is the weight of the transitive closure structure and we uniformly select it from  $\{0.1, 0.01\}$  here. Generally, a sparser network implies that incorporating the transitive closure structure is more important, so we can set 0.1 for the sparser network and 0.01 for the rest.

Finally, we have the following overall objective function:

$$L = \min_{\mathbf{U}, \mathbf{V}, \mathbf{S}} \|\mathbf{A} - \mathbf{UVU}^T - \mathbf{S}\|_F^2 + \lambda_1 \|\mathbf{U}\|_F^2 + \lambda_2 \|\mathbf{V}\|_F^2 + \lambda_3 \|\mathbf{S}\|_F^2 + \lambda_4 \|\mathbf{S}\|_1. \quad (4)$$

By optimizing model (4), we can obtain the optimal  $\mathbf{U}$ ,  $\mathbf{V}$ , and  $\mathbf{S}$ . The potential trust relations can be inferred based on  $\mathbf{UVU}^T + \mathbf{S}$ .

### 4 Optimization

The objective function (4) is not convex, and to solve it efficiently, a common strategy is to use an alternating minimiza-

tion approach, which separates the optimization of (4) to three subproblems.

**U-subproblem:** The partial derivation of (4) with respect to  $\mathbf{U}$  is as follows:

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{U}} = & -2(\mathbf{A} - \mathbf{S})\mathbf{U}\mathbf{V}^T - 2(\mathbf{A} - \mathbf{S})^T\mathbf{U}\mathbf{V} \\ & + 2\mathbf{U}\mathbf{V}\mathbf{U}^T\mathbf{U}\mathbf{V}^T + 2\mathbf{U}\mathbf{V}^T\mathbf{U}^T\mathbf{U}\mathbf{V} + 2\lambda_1\mathbf{U}. \end{aligned} \quad (5)$$

So according to gradient descent method,  $\mathbf{U}$  can be updated as:

$$\mathbf{U} \leftarrow \mathbf{U} - \gamma_u \frac{\partial L_U}{\partial \mathbf{U}}, \quad (6)$$

where  $\gamma_u > 0$  is the learning step.

**V-subproblem:** The partial derivation of (4) with respect to  $\mathbf{V}$  is as follows:

$$\frac{\partial L}{\partial \mathbf{V}} = -2\mathbf{U}^T(\mathbf{A} - \mathbf{S})\mathbf{U} + 2\mathbf{U}^T\mathbf{U}\mathbf{V}\mathbf{U}^T\mathbf{U} + 2\lambda_2\mathbf{V}. \quad (7)$$

Similarly,  $\mathbf{V}$  can be updated by gradient descent method with learning step  $\gamma_v > 0$ :

$$\mathbf{V} \leftarrow \mathbf{V} - \gamma_v \frac{\partial L_V}{\partial \mathbf{V}}. \quad (8)$$

**S-subproblem:** updating  $\mathbf{S}$  with other parameters in (4) fixed needs to solve the following function:

$$L_S = \min_{\mathbf{S}} \|\mathbf{A} - \mathbf{U}\mathbf{V}\mathbf{U}^T - \mathbf{S}\|_F^2 + \lambda_3\|\mathbf{S}\|_F^2 + \lambda_4\|\mathbf{S}\|_1. \quad (9)$$

By introducing another auxiliary variable  $\mathbf{M} = \mathbf{S}$ , the function (9) can be transformed as:

$$\begin{aligned} L_S = \min_{\mathbf{S}, \mathbf{M}, \mathbf{P}} & \|\mathbf{A} - \mathbf{U}\mathbf{V}\mathbf{U}^T - \mathbf{S}\|_F^2 + \lambda_3\|\mathbf{S}\|_F^2 + \lambda_4\|\mathbf{M}\|_1 \\ & + \lambda_5\|\mathbf{S} - \mathbf{M} + \mathbf{P}\|_F^2, \end{aligned} \quad (10)$$

where  $\lambda_5 > 0$  is the penalty parameter, and  $\mathbf{P}$  is the scaled dual variable. The optimization of (10) can be further divided into the following three parts:

(1) Fix  $\mathbf{S}$ ,  $\mathbf{P}$ , and update  $\mathbf{M}$ :

$$\mathbf{M} = \mathcal{S}\left(\mathbf{S} + \mathbf{P}, \frac{\lambda_4}{\lambda_5}\right), \quad (11)$$

where  $\mathcal{S}(a, b) = \text{sign}(a)\max(|a| - b, 0)$  and  $\text{sign}(x)$  is the signum function, i.e., if  $x > 0$ ,  $\text{sign}(x) = 1$ ; if  $x < 0$ ,  $\text{sign}(x) = -1$ ; otherwise,  $\text{sign}(x) = 0$ .

(2) Fix  $\mathbf{M}$ ,  $\mathbf{P}$ , and update  $\mathbf{S}$ :

By setting the partial derivation of (10) with respect to  $\mathbf{S}$  to 0,  $\mathbf{S}$  can be updated by

$$\mathbf{S} \leftarrow [\mathbf{A} - \mathbf{U}\mathbf{V}\mathbf{U}^T + \lambda_5(\mathbf{M} - \mathbf{P})]/(1 + \lambda_3 + \lambda_5). \quad (12)$$

(3) Fix  $\mathbf{S}$ ,  $\mathbf{M}$ , and update  $\mathbf{P}$ :

$$\mathbf{P} \leftarrow \mathbf{P} + \mathbf{S} - \mathbf{M}. \quad (13)$$

Networks	Advogato	Ciao	Epinions
Users	6541	7375	8519
Trust relations	51127	111781	300091
Average in/out-degree	7.8164	15.1567	35.2261
Max in-degree	722	100	1706
Max out-degree	786	804	1303
Sparsity	0.1195%	0.2055%	0.4135%

Table 1: Description of three networks.

## 4.1 Convergence Issue

Although the convergence to a local minimum is difficult to be guaranteed, we empirically show that our proposed optimization has a very strong convergence behavior, shown in Section 5. We give a weak convergence result that under some mild conditions, any limit point of the iteration sequence generated by the updating rule is a Karush-Kuhn-Tucker (KKT) point. It is worth proving that any converging point must be a KKT point because it is a necessary condition to be a local optimal solution.

Our objective function (4) can be rewritten as:

$$\begin{aligned} L = \min_{\mathbf{U}, \mathbf{V}, \mathbf{S}, \mathbf{M}} & \|\mathbf{A} - \mathbf{U}\mathbf{V}\mathbf{U}^T - \mathbf{S}\|_F^2 + \lambda_1\|\mathbf{U}\|_F^2 + \lambda_2\|\mathbf{V}\|_F^2 \\ & + \lambda_3\|\mathbf{S}\|_F^2 + \lambda_4\|\mathbf{M}\|_1, \\ \text{s.t. } & \mathbf{S} = \mathbf{M}. \end{aligned} \quad (14)$$

One KKT condition of (14) can be derived as  $\mathbf{\Lambda} \in \lambda_4\partial_{\mathbf{M}}(\|\mathbf{M}\|_1)$ , where  $\mathbf{\Lambda}$  are the Lagrange multipliers, which is equivalent to

$$\begin{aligned} \mathbf{S} + \frac{\mathbf{\Lambda}}{\lambda_5} & \in \mathbf{S} + \frac{\lambda_4}{\lambda_5}\partial_{\mathbf{M}}(\|\mathbf{M}\|_1) \\ & = \mathbf{M} + \frac{\lambda_4}{\lambda_5}\partial_{\mathbf{M}}(\|\mathbf{M}\|_1) \triangleq \mathcal{Q}_{\frac{\lambda_5}{\lambda_4}}(\mathbf{M}), \end{aligned} \quad (15)$$

where the scalar function  $\mathcal{Q}_\beta(t) \triangleq \frac{1}{\beta}|\partial t| + t$  is applied element-wise to  $\mathbf{M}$ . According to [Shen *et al.*, 2014],  $\mathcal{Q}_\beta(t)$  is monotone so that  $\mathcal{Q}_\beta^{-1}(t) \triangleq \mathcal{S}(t, \frac{1}{\beta})$ . By applying  $\mathcal{Q}_\beta^{-1}(\cdot)$  to (15), we have the following relation:

$$\mathbf{M} = \mathcal{Q}_{\frac{\lambda_5}{\lambda_4}}^{-1}\left(\mathbf{S} + \frac{\mathbf{\Lambda}}{\lambda_5}\right) \equiv \mathcal{S}\left(\mathbf{S} + \frac{\mathbf{\Lambda}}{\lambda_5}, \frac{\lambda_4}{\lambda_5}\right). \quad (16)$$

Finally, the KKT conditions for (14) can be written as:

$$\begin{aligned} \mathbf{S} - \mathbf{M} = 0, \quad \frac{\partial L}{\partial \mathbf{U}} = 0, \quad \frac{\partial L}{\partial \mathbf{V}} = 0, \\ \frac{\partial L}{\partial \mathbf{S}} = 0, \quad \mathbf{M} = \mathcal{S}\left(\mathbf{S} + \frac{\mathbf{\Lambda}}{\lambda_5}, \frac{\lambda_4}{\lambda_5}\right). \end{aligned} \quad (17)$$

Then based on these conditions, we have the following theorem:

**Theorem 1.** Let  $\mathbf{X} \triangleq (\mathbf{U}, \mathbf{V}, \mathbf{S}, \mathbf{M}, \mathbf{P})$  and  $\{\mathbf{X}^j\}_{j=1}^\infty$  be generated by the proposed optimization algorithm. Assume that  $\{\mathbf{X}^j\}_{j=1}^\infty$  is bounded and  $\lim_{j \rightarrow \infty} (\mathbf{X}^{j+1} - \mathbf{X}^j) = 0$ . Then any accumulation point of  $\{\mathbf{X}^j\}_{j=1}^\infty$  satisfies the KKT conditions (17). In particular, whenever  $\{\mathbf{X}^j\}_{j=1}^\infty$  converges, it converges to KKT point of (14).

Networks	Training set	Random	CN	TP	SVD	RRMC	sTrust	BNMF	SNMF	HNMF	Ours
Advogato	50%	0.057	11.09	16.63	13.37	4.562	13.71	18.50	17.29	19.64	<b>20.08*</b>
	60%	0.044	10.90	15.67	14.17	5.950	14.36	18.20	18.30	18.69	<b>19.46*</b>
	70%	0.036	10.47	14.12	14.64	5.613	14.13	17.46	17.87	17.62	<b>18.32*</b>
	80%	0.020	9.055	11.99	14.04	4.488	13.09	15.73	16.87	16.83	<b>17.32*</b>
	90%	0.004	6.928	8.569	12.22	4.096	10.81	13.27	13.86	13.23	<b>14.21*</b>
Ciao	50%	0.104	16.78	18.54	13.66	4.493	17.37	18.41	18.58	20.49	<b>21.69*</b>
	60%	0.078	15.48	16.85	13.37	4.460	17.12	18.66	18.64	18.79	<b>20.12*</b>
	70%	0.058	13.56	14.82	12.59	3.830	16.02	17.84	17.40	18.23	<b>18.60*</b>
	80%	0.043	11.09	12.19	11.01	3.094	13.91	15.39	15.51	15.86	<b>16.08**</b>
	90%	0.011	7.224	8.284	8.088	2.319	10.25	11.17	11.71	11.22	<b>12.13**</b>
Epinions	50%	0.212	18.01	19.42	19.16	7.750	22.40	23.04	23.22	24.53	<b>25.16*</b>
	60%	0.173	16.82	17.89	19.18	7.592	21.61	23.16	23.33	23.94	<b>24.28*</b>
	70%	0.124	15.07	16.02	18.39	7.322	19.93	20.58	22.53	21.18	<b>22.90*</b>
	80%	0.086	12.68	13.61	16.65	7.063	17.35	16.63	20.48	17.08	<b>20.89*</b>
	90%	0.031	9.213	10.18	13.35	6.725	13.23	13.12	16.45	12.69	<b>16.68*</b>

Table 2: Accuracies (%) of the trust prediction models (significantly outperforms the baselines at the: \*0.01 level and \*\*0.05 level)

*Proof.* First, we can denote  $\mathbf{P} = \frac{\mathbf{A}}{\lambda_5}$ . From the updating rule (13), we have  $\mathbf{P}_+ = \mathbf{P} + \mathbf{S} - \mathbf{M}$ , where  $\mathbf{P}_+$  is a next point of  $\mathbf{P}$  in a sequence  $\{\mathbf{P}^j\}_{j=1}^{\infty}$ . If sequence of variables  $\{\mathbf{P}^j\}_{j=1}^{\infty}$  converges to a stationary point, i.e.,  $(\mathbf{P}_+ - \mathbf{P}) \rightarrow 0$ , then  $(\mathbf{S} - \mathbf{M}) \rightarrow 0$ , which satisfies the first condition of KKT conditions.

Second, assume  $\mathbf{U}_+$  is a next point of  $\mathbf{U}$  derived in (6), due to  $\gamma_u > 0$ , if  $(\mathbf{U}_+ - \mathbf{U}) \rightarrow 0$ , then it is easy to verify that the second condition in (17) is satisfied.

Likewise, it is easy to check that the third condition in (17) is satisfied.

Fourth, from  $\mathbf{S}_+$  derived in the algorithm, we have

$$\mathbf{S}_+ - \mathbf{S} = [\mathbf{A} - \mathbf{UVU}^T + \lambda_5(\mathbf{M} - \mathbf{P})]/(1 + \lambda_3 + \lambda_5) - \mathbf{S}. \quad (18)$$

By multiplying  $(1 + \lambda_3 + \lambda_5)$  to both sides in (18), it can be written as

$$(\mathbf{S}_+ - \mathbf{S})(1 + \lambda_3 + \lambda_5) = \mathbf{A} - \mathbf{UVU}^T - \mathbf{S} - \lambda_3\mathbf{S} - \lambda_5(\mathbf{S} - \mathbf{M} + \mathbf{P}). \quad (19)$$

So when  $(\mathbf{S}_+ - \mathbf{S}) \rightarrow 0$ , we can derive  $[\mathbf{A} - \mathbf{UVU}^T - \mathbf{S} - \lambda_3\mathbf{S} - \lambda_5(\mathbf{S} - \mathbf{M} + \mathbf{P})] \rightarrow 0$ , i.e., the fourth condition is satisfied.

Finally, from (11), we have  $\mathbf{M}_+ - \mathbf{M} = \mathcal{S}(\mathbf{S} + \mathbf{P}, \frac{\lambda_4}{\lambda_5}) - \mathbf{M}$ . So if  $(\mathbf{M}_+ - \mathbf{M}) \rightarrow 0$ , then  $[\mathcal{S}(\mathbf{S} + \mathbf{P}, \frac{\lambda_4}{\lambda_5}) - \mathbf{M}] \rightarrow 0$ , which satisfies the last condition. Therefore, the sequence  $\{\mathbf{X}^j\}_{j=1}^{\infty}$  asymptotically satisfies the KKT conditions for (14), which completes the proof.  $\square$

Since  $\mathbf{A}$  and  $\mathbf{S}$  are very sparse, we denote  $N_A$  and  $N_S$  as the number of non-zeros in  $\mathbf{A}$  and  $\mathbf{S}$ , respectively. The computations of updating rules in (5), (7), (11), (12), and (13) are  $\mathcal{O}(N_A k + N_S k + nk^2)$ ,  $\mathcal{O}(N_A k + N_S k + nk^2)$ ,  $\mathcal{O}(n^2)$ ,  $\mathcal{O}(nk^2 + n^2 k)$ , and  $\mathcal{O}(n^2)$ , respectively. Since usually  $k \ll n$ , consequently, the overall computation of the updating rules is  $\mathcal{O}(n^2 k)$ .

## 5 Experimental Evaluations

We employed the following three real world trust networks for the evaluations: **Advogato** [Massa *et al.*, 2009], **Ciao**

[Tang and Liu, 2015], and **Epinions** [Tang and Liu, 2015]. A detailed summary of the networks used in our experiments is presented in Table 1.

We compared our method against the following nine trust prediction models which only consider network structure. **Random**: this baseline randomly selects trust relations among the pairs of users. **CN** [Liben-Nowell and Kleinberg, 2003] and **TP** [Guha *et al.*, 2004] are based on the neighborhood structure and propagation, respectively; **SVD** [Abdi, 2007], **RRMC** [Huang *et al.*, 2013], **sTrust** [Wang *et al.*, 2015], **BNMF**, **SNMF**, and **HNMF** are all low-rank based trust prediction models. BNMF is the basic model in Eq.(2), SNMF and HNMF are the basic models in Eq.(2) with only the sparsity term and the high-order structure, respectively.

### 5.1 Experimental Results

To obtain the best possible performance of the compared methods, we tuned their corresponding parameters. For the low rank based methods, we uniformly set the rank  $k = 100$  for all the networks. For the parameters in our model, we also uniformly tuned  $\lambda_1, \lambda_2, \lambda_3, \lambda_4$  from  $\{0.01, 0.1, 1, 10, 100\}$ . To simplify and shrink the parameter space, we let  $\lambda_1 = \lambda_2$  and  $\lambda_5 = 100$ , although the simplification may very likely exclude the best performance of our method.

We followed the widely used metric for unsupervised trust prediction to evaluate all these methods [Wang *et al.*, 2015]. Please note that only the adjacency matrix  $\mathbf{A}$  is the observed trust relations, so the prediction accuracy is calculated on  $\mathbf{A}$ . Specifically, we divided a network into two parts  $\mathcal{B}$  and  $\mathcal{H}$ , where  $\mathcal{B}$  is the set of user pairs with trust relations and  $\mathcal{H}$  is the set of user pairs without trust relations. We then randomly chose  $x\%$  of  $\mathcal{B}$  as the training set  $\mathcal{Q}$  and the remaining  $1 - x\%$  of  $\mathcal{B}$  as the testing set  $\mathcal{N}$  to be predicted. Based on  $\mathcal{Q}$ , we can predict the trust scores of user pairs in  $\mathcal{N} \cup \mathcal{H}$ . Then we ranked them in a descending order and selected the top- $|\mathcal{N}|$  ranked pairs as the predicted trust relations  $\mathcal{C}$ . The accuracy of the predicted trust relations is  $AC = \frac{|\mathcal{N} \cap \mathcal{C}|}{|\mathcal{N}|}$ . To comprehensively assess the performance, we varied  $x$  from  $\{50, 60, 70, 80, 90\}$ . The whole procedure was repeated 5 times, and the average accuracy was reported. Please note

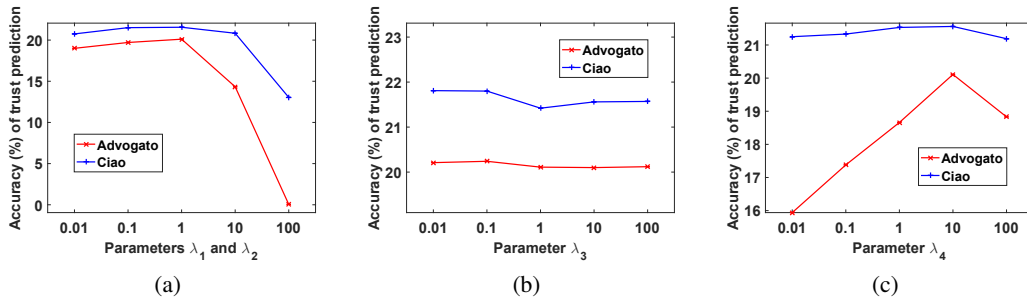


Figure 3: The effect of parameters  $\lambda_1 = \lambda_2$ ,  $\lambda_3$ , and  $\lambda_4$  on Advogato and Ciao.

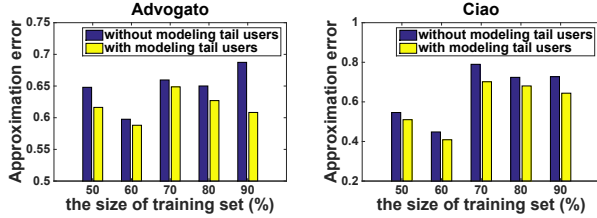


Figure 4: Approximation error analysis.

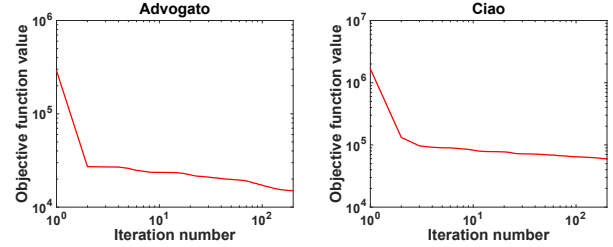


Figure 5: Convergence analysis.

that RRMC is a matrix completion based method, so the observed variables need to be explicitly specified. For a fair comparison, we also selected  $x\%$  of  $\mathcal{H}$ , together with  $\mathcal{N}$ , as the training set, and predicted the trust relations of the rest user pairs based on ranking. Otherwise, providing full  $\mathcal{H}$  will result in a trivial solution of nearly 100% accuracy since all zeros are already known by RRMC.

The performance of different trust prediction methods is shown in Table 2. As can be seen, our proposed method achieves significant improvements in terms of accuracy over all the tested networks, which indicates the superiority of our method. Moreover, SNMF and HNMF are generally better than BNMF, which indicates the effectiveness of incorporating the sparsity term and the high-order structure. Although TP considers more structure information, SVD and BNMF still achieve competitive results, indicating the potential of low-rank based method. Further, by integrating additional domain knowledge with matrix factorization, sTrust achieves better performance than topology based methods (CN and TP). As for RRMC, its low accuracy is probably because that both high-order structure and tail users are not considered; also, it only utilizes the partial trust links and partial unobserved trust links in the training procedure, in order to avoid the trivial solution mentioned before.

## 5.2 Parameter Analysis

Here we tested the effect of parameters  $\lambda_1$ ,  $\lambda_2$ ,  $\lambda_3$  and  $\lambda_4$  of our model on the real trust networks. Because the results of different networks show similar trends, we just used two networks (Advogato and Ciao) as examples here, shown in Figure.3. In (a), the accuracy usually increases first and then falls down again, suggesting that these two parameters cannot be too small or too large. The largest accuracy usually appears when  $\lambda_1$  and  $\lambda_2$  are around 1. As for  $\lambda_3$  in (b), the

accuracy of trust prediction is relatively stable. With different values of  $\lambda_3$ , the change of accuracy is not too drastic, suggesting that  $\lambda_3$  is easy to be set within this range. In (c), we can observe that the accuracy is low with a small value of  $\lambda_4$ . After that, the accuracy increases and finally achieves the highest value when  $\lambda_4$  is around 10. This demonstrates the importance of modeling tail users in  $\mathbf{S}$ . But still, too large values of  $\lambda_4$  will result in a low accuracy.

## 5.3 Approximation Error Analysis

Here, we further analyzed the approximation errors on different training sets. In particular, when we did not consider the tail users, we calculated the relative approximation errors as  $\|\mathbf{A} - \mathbf{UVU}^T\|_F^2 / \|\mathbf{A}\|_F^2$ ; when the tail users were modeled, we calculated its relative approximation errors as  $\|\mathbf{A} - \mathbf{UVU}^T - \mathbf{S}\|_F^2 / \|\mathbf{A}\|_F^2$ . All the errors were recorded when the best performance in trust prediction was achieved. The results are shown in Figure. 4. As is clear, for all the cases, the approximation errors with modeling tail users are consistently smaller than those without considering the tail users. This empirically verifies that modeling tail users is able to help better approximate the trust matrix, alleviating the problem of the violated low-rank assumption.

Finally, we empirically analyzed its convergence property, shown in Figure.5. It can be seen that the objective function values are non-increasing and drop sharply within a small number of iterations (about 5 iterations), suggesting that the proposed algorithm has a strong convergence behavior.

## 6 Conclusion

We studied the problem of modeling the power-law distribution for trust prediction under the framework of low-rank matrix factorization. The extensive experimental results and

the related analysis indicate the superiority of the proposed method. The tail users widely exist in many networks, which makes them an important factor that cannot be ignored for predicting the trusts. It would be interesting to study more powerful trust prediction methods under this framework.

## Acknowledgments

This work was supported in part by National Program on Key Basic Research Project (No. 2015CB352300), National Natural Science Foundation of China (No. 61702296, No. 61772304, No. 61531006, No. 61502334), the research fund of Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology, and the Young Elite Scientist Sponsorship Program by CAST.

## References

- [Abdi, 2007] Hervé Abdi. Singular value decomposition (svd) and generalized singular value decomposition. *Encyclopedia of measurement and statistics. Thousand Oaks (CA): Sage*, pages 907–12, 2007.
- [Beigi *et al.*, 2016] Ghazaleh Beigi, Jiliang Tang, Suhang Wang, and Huan Liu. Exploiting emotional information for trust/distrust prediction. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 81–89. SIAM, 2016.
- [Guha *et al.*, 2004] Ramanathan Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. Propagation of trust and distrust. In *Proceedings of the 13th international conference on World Wide Web*, pages 403–412. ACM, 2004.
- [Hendriks *et al.*, 2015] Ferry Hendriks, Kris Bubendorfer, and Ryan Chard. Reputation systems: A survey and taxonomy. *Journal of Parallel and Distributed Computing*, 75:184–197, 2015.
- [Huang *et al.*, 2013] Jin Huang, Feiping Nie, Heng Huang, Yu Lei, and Chris HQ Ding. Social trust prediction using rank-k matrix recovery. In *IJCAI*, pages 2647–2653, 2013.
- [Josang and Ismail, 2002] Audun Josang and Roslan Ismail. The beta reputation system. In *Proceedings of the 15th bled electronic commerce conference*, volume 5, pages 2502–2511, 2002.
- [Liben-Nowell and Kleinberg, 2003] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. In *Conference on Information and Knowledge Management*, pages 556–559, 2003.
- [Massa *et al.*, 2009] Paolo Massa, Martino Salvetti, and Danilo Tomasoni. Bowling alone and trust decline in social network sites. In *Dependable, Autonomic and Secure Computing, 2009. DASC’09. Eighth IEEE International Conference on*, pages 658–663. IEEE, 2009.
- [O’Donovan *et al.*, 2007] John O’Donovan, Barry Smyth, Vesile Evrim, and Dennis McLeod. Extracting and visualizing trust relationships from online auction feedback comments. In *IJCAI*, pages 2826–2831, 2007.
- [Shen *et al.*, 2014] Yuan Shen, Zaiwen Wen, and Yin Zhang. Augmented lagrangian alternating direction method for matrix separation based on low-rank factorization. *Optimization Methods and Software*, 29(2):239–263, 2014.
- [Sherchan *et al.*, 2013] Wanita Sherchan, Surya Nepal, and Cecile Paris. A survey of trust in social networks. *ACM Computing Surveys (CSUR)*, 45(4):47, 2013.
- [Tang and Liu, 2015] Jiliang Tang and Huan Liu. Trust in social media. *Synthesis Lectures on Information Security, Privacy, & Trust*, 10(1):1–129, 2015.
- [Tang *et al.*, 2012] Jiliang Tang, Huiji Gao, and Huan Liu. mtrust: discerning multi-faceted trust in a connected world. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 93–102. ACM, 2012.
- [Tang *et al.*, 2013] Jiliang Tang, Huiji Gao, Xia Hu, and Huan Liu. Exploiting homophily effect for trust prediction. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 53–62. ACM, 2013.
- [Wang *et al.*, 2015] Ying Wang, Xin Wang, Jiliang Tang, Wanli Zuo, and Guoyong Cai. Modeling status theory in trust prediction. In *AAAI*, pages 1875–1881, 2015.
- [Wang *et al.*, 2017a] Jing Wang, Jie Shen, Ping Li, and Huan Xu. Online matrix completion for signed link prediction. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 475–484. ACM, 2017.
- [Wang *et al.*, 2017b] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. Community preserving network embedding. In *AAAI*, pages 203–209, 2017.
- [Xiong and Liu, 2004] Li Xiong and Ling Liu. Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE transactions on Knowledge and Data Engineering*, 16(7):843–857, 2004.
- [Xu *et al.*, 2016] Chang Xu, Dacheng Tao, and Chao Xu. Robust extreme multi-label learning. In *KDD*, pages 1275–1284, 2016.
- [Yao *et al.*, 2014] Yuan Yao, Hanghang Tong, Xifeng Yan, Feng Xu, and Jian Lu. Multi-aspect+ transitivity+ bias: An integral trust inference model. *IEEE Transactions on Knowledge and Data Engineering*, 26(7):1706–1719, 2014.
- [Zheng *et al.*, 2014] Xiaoming Zheng, Yan Wang, Mehmet A Orgun, Youliang Zhong, Guanfeng Liu, et al. Trust prediction with propagation and similarity regularization. In *AAAI*, pages 237–244, 2014.
- [Zhou and Hwang, 2007] Runfang Zhou and Kai Hwang. Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing. *IEEE Transactions on parallel and distributed systems*, 18(4), 2007.
- [Zou and Hastie, 2005] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.