Wang Y, Wang X, Zuo WL. Research on trust prediction from a sociological perspective. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 30(4): 843–858 July 2015. DOI 10.1007/s11390-015-1564-8

Research on Trust Prediction from a Sociological Perspective

Ying Wang [1,2] (王 英), *Senior Member, CCF, ACM*, Xin Wang [1,2,3,*] (王 鑫), *Member, CCF, ACM*, and Wan-Li Zuo [1,2] (左万利), *Senior Member, CCF*

[1] *College of Computer Science and Technology, Jilin University, Changchun 130012, China*

[2] *Key Laboratory of Symbolic Computation and Knowledge Engineering, Ministry of Education, Changchun 130012, China*

[3] *School of Computer Technology and Engineering, Changchun Institute of Technology, Changchun 130012, China*

E-mail: wangying2010@jlu.edu.cn; wangxccs@126.com; zuowl@jlu.edu.cn

Received January 28, 2015; revised April 29, 2015.

Abstract Trust, as a major part of human interactions, plays an important role in helping users collect reliable information and make decisions. However, in reality, user-specified trust relations are often very sparse and follow a power law distribution; hence inferring unknown trust relations attracts increasing attention in recent years. Social theories are frameworks of empirical evidence used to study and interpret social phenomena from a sociological perspective, while social networks reflect the correlations of users in real world; hence, making the principle, rules, ideas and methods of social theories into the analysis of social networks brings new opportunities for trust prediction. In this paper, we investigate how to exploit homophily and social status in trust prediction by modeling social theories. We first give several methods to compute homophily coefficient and status coefficient, then provide a principled way to model trust prediction mathematically, and propose a novel framework, hsTrust, which incorporates homophily theory and status theory. Experimental results on real-world datasets demonstrate the effectiveness of the proposed framework. Further experiments are conducted to understand the importance of homophily theory and status theory in trust prediction.

Keywords trust prediction, homophily coefficient, status coefficient, social theory, matrix factorization

1 Introduction

With the pervasiveness of social media, more and more users participate in various online activities and produce data in an unprecedented rate. Users are both producers and consumers of data so that it is vital to provide a satisfactory trust prediction model which resolves information overload, increased uncertainties and risk from unreliable information. Trust prediction, which explores unknown relations between online users, is an emerging and important research topic in social network analysis and many web applications in recent years, such as trust-aware recommendation systems[1-2], finding high-quality user generated content[3-4], and viral marketing[5-6]. However, in reality, the available explicit trust relations are extremely sparse, and follow a power law distribution, suggesting a great challenge to trust prediction. Although many theoretical models and systems have been developed on trust prediction[7-8], there are still some limitations and little work exploiting social theories for trust prediction.

Social theories are frameworks of empirical evidence used to study and interpret social phenomena from a sociological perspective. There are many social theories developed from social sciences to explain social phenomena, i.e., the homophily theory suggests how individuals connect to each other, the balance theory conceptualizes the cognitive consistency motive as a drive toward psychological balance, and the status theory refers to the position or rank of a user. Recent

Regular Paper

Special Section on Data Management and Data Mining

This work is supported by the National Natural Science Foundation of China under Grant No. 61300148, the Scientific and Technological Break-Through Program of Jilin Province of China under Grant No. 20130206051GX, and the Science and Technology Development Program of Jilin Province of China under Grant No. 20130522112JH.

*Corresponding Author

advances in computer science community provide necessary computational tools and techniques for us to verify social theories on large-scale media data, e.g., for the homophily theory, users in Epinions with trust relations are likely to rate same items with similar scores[9]; for the balance theory, users in a social network tend to be formed into a balanced network structure[10-11]; for the status theory, in [12], it is reported that 99% of triads in the Enron email social network and the advisor-advisee social network satisfy status theory, and similar patterns are observed on Epinions and Wikipedia datasets in [13]. With the verification of more and more social theories in social media data, integrating social theories with computational models becomes an appealing direction to mine trust prediction.

In this paper, we mainly focus on trust prediction by exploring social theories. In essence, we investigate how to model the homophily theory and the status theory mathematically, and how to incorporate them for trust prediction, which result in a novel unsupervised framework, hsTrust. Our major contributions include:

- an approach to modeling the homophily theory in trust relations mathematically via homophily regularization;
- an approach to modeling the status theory in trust relations mathematically via status regularization;
- an unsupervised framework, hsTrust, to predict trust relations between users based on the combination of the homophily theory and the status theory;
- experimental results on real-world datasets from Epinions and Ciao demonstrating the effectiveness of hsTrust, and elaborating the importance of the homophily theory and the status theory in trust prediction.

The rest of the paper is organized as follows. Section 2 gives the problem statement for trust prediction. Section 3 gives a brief introduction on social theories. Section 4 proposes an unsupervised framework, hsTrust, to predict trust relations. Section 5 reports experimental results on real-world datasets with discussions. Section 6 outlines the background and related work on trust prediction. Section 7 finally concludes and presents the future work.

2 Problem Statement

In this section, we first introduce the notations used in the paper and then formally define the problem we study.

Notations. Let $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$ be the set of users where n is the number of users. $\boldsymbol{P}$ denotes the user-rating matrix about items, and each element $P(i, j)$ is the rating to the j-th item from u_i. $\boldsymbol{Z}$ is the homophily coefficient matrix among n users, and each element $\zeta(i, j)$ denotes the homophily coefficient between u_i and u_j. $\boldsymbol{R}$ is the status coefficient matrix among n users, and each element $\eta(i, j)$ denotes the status coefficient between u_i and u_j. $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ is the set of status scores for $\mathcal{U}$ where s_i denotes the status score of u_i, and $s_i \in [0, 1]$. The larger s_i is, the higher the status of u_i is. We further assume that users in $\mathcal{U}$ are sorted by their status scores $\mathcal{S}$ in descending order. That is to say, for u_i and u_j, if $i < j$, then $s_i > s_j$. We use $\boldsymbol{G} \in \mathbb{R}^{n\times n}$ to denote user-user trust relations where $G_{ij} = 1$ if u_i trusts u_j, zero otherwise. For a matrix named $\boldsymbol{A}$, $\boldsymbol{A}^{\mathrm{T}}$ denotes the transpose of matrix $\boldsymbol{A}$, $\|\boldsymbol{A}\|$ denotes the Euclidean norm, and $\|\boldsymbol{A}\|_{\mathrm{F}}$ denotes the Frobenius norm of $\boldsymbol{A}$, specifically, $\|\boldsymbol{A}\|_{\mathrm{F}} = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} \boldsymbol{A}_{ij}^2}$.

With the given notations, we formally define the problem of predicting trust relations through exploring social theories from a social perspective as: given a set of users $\mathcal{U}$ with social network information $\boldsymbol{G}$, status score set $\mathcal{S}$ and user-rating matrix $\boldsymbol{P}$, we aim to obtain homophily coefficient matrix $\boldsymbol{Z}$ and status coefficient matrix $\boldsymbol{R}$, and then identify the likelihood of u_i and u_j to establish trust relation by exploring social theories.

3 Social Theories in Trust Relations

Social theories help bridge the gap from what we have to what we want to understand in social media. There are many social theories developed to explain various types of social phenomena, thereinto, the homophily theory, the balance theory and the social theory are the most common theories in mining social media data. Specially, the balance theory is generally intended as a model for undirected networks, while the status theory is developed for directed social networks. In this paper, we investigate the problem of trust prediction in directed social networks; therefore, we only focus on the homophily theory and the status theory.

3.1 Homophily

3.1.1 Homophily Theory

The homophily theory is the tendency of individuals to associate with those similar to themselves, and has been documented across a wide array of different characteristics, including race, age, ethnicity, profession, religion, various behaviors, etc. Fig.1 demonstrates the homophily theory that users are more likely to establish trust relations with users with similar preferences than those users without similar preferences; users are more likely to establish colleague relations with users working in the same place than those users working in different places. Homophily can be one of the most pervasive and robust tendencies of social networks, and one of the main features that makes users distinctively social.

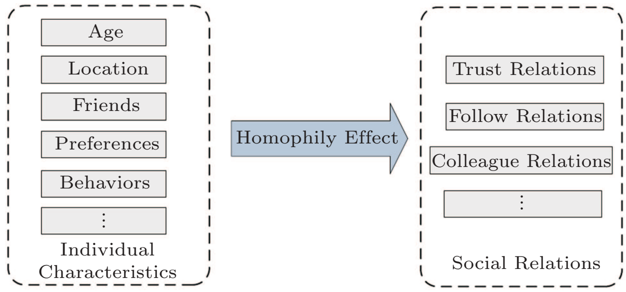


Fig.1. Homophiliy theory.

The homophily theory indicates that users with similar tastes are more likely to be socially connected. For trust prediction, the homophily effect suggests that similar users have a higher likelihood to establish trust relations. For example, people with similar tastes about items are more likely to trust each other in product review sites. Exploiting homophily effect provides a new perspective for trust prediction.

3.1.2 *Homophily Coefficient*

We investigate homophily via studying users' similarity. Let $\boldsymbol{Z}$ be the homophily coefficient matrix among n users, defined as:

$$\boldsymbol{Z} = \begin{pmatrix} \zeta(1,1) & \zeta(1,2) & \cdots & \zeta(1,n) \\ \zeta(2,1) & \zeta(2,2) & \cdots & \zeta(2,n) \\ \vdots & \vdots & \ddots & \vdots \\ \zeta(n,1) & \zeta(n,2) & \cdots & \zeta(n,n) \end{pmatrix},$$

and each element $\zeta(i,j)$ denotes the homophily coefficient between u_i and u_j. In the context of product view sites, the user preference can be inferred from the user's ratings; hence homophily coefficient in this work is simply measured via rating although there are other more sophisticated measures[14]. For u_i, we assume that $I(i)$ is the set of items u_i rates and P_{ij} is the rating to the j-th item from u_i. We investigate the following three widely used similarity measures[15] for homophily coefficient.

• *Jaccard's Coefficient* (*JC*). Jaccard's coefficient is defined as the number of common rated items of two users divided by the total number of their unique rated items, formally stated as:

$$\zeta(i,j) = JC(u_i,u_j) = \frac{|I(i) \cap I(j)|}{|I(i) \cup I(j)|}.$$

• *Rating Similarity* (*RS*). *JC* counts the common rated items; however, different users might rate the same item differently. For example, u_i rates the j-th item as 5 stars while u_j gives 1 star to the j-th item. To capture different tastes from different users, we define the rating similarity *RS* as:

$$\zeta(i,j) = RS(u_i,u_j) = \frac{\sum_k P_{ik} \times P_{jk}}{\sqrt{\sum_k P_{ik}^2}\sqrt{\sum_k P_{jk}^2}},$$

and actually, $RS(u_i,u_j)$ is the cosine similarity between the rating vectors of u_i and u_j.

• *Pearson Correlation Coefficient* (*PCC*). Different users may have different rating styles: some users have the propensity to give higher ratings to all items while others probably tend to rate lowly, motivating us to propose *PCC*:

$$\begin{aligned} \zeta(i,j) &= PCC(u_i,u_j) \\ &= \frac{\sum_{k \in I(i) \cap I(j)} (P_{ik} - \bar{P}_i) \times (P_{jk} - \bar{P}_j)}{\sqrt{\sum_k (P_{ik} - \bar{P}_i)^2}\sqrt{\sum_k (P_{jk} - \bar{P}_j)^2}}, \end{aligned}$$

where $\bar{P}_i$ denotes the average rate of u_i and k belongs to the subset of items rated by both u_i and u_j.

3.2 Social Status

3.2.1 *Status Theory*

Social status is an important concept in trust, which refers to the position or rank of a user in a social community, and represents the degree of honor or prestige attached to the position of each individual[16]. The status theory only makes sense with directed links, since it posits a status differential from the creator of a link to its recipient. In the status theory, we consider a positive directed link to indicate that the creator of the link regards the recipient as having higher status, and a negative directed link indicates that the recipient is viewed as having lower status. These relative levels of status can then be propagated along multi-step paths of signed links, often leading to different predictions[13,17]. Fig.2 shows the trust propagation based on the status theory, i.e., Fig.2(a) shows this situation in which u_1 links positively to u_2, and u_2 in turn links positively to u_3, namely, the status theory predicts that u_1 regards

u_2 as having higher status, and u_2 regards u_3 as having higher status, so u_3 should regard u_1 as having low status and hence be inclined to link negatively to u_1, that is to say, a trust relation is more likely to be established from u_1 to u_3 than from u_3 to u_1.

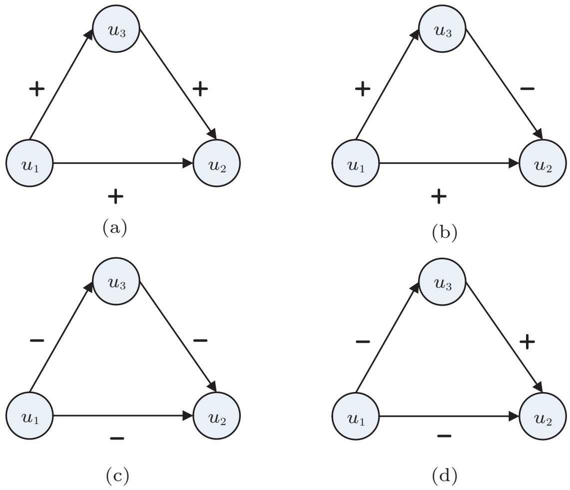


Fig.2. Status theory.

The status theory is developed to explain how users trust each other based on their statuses, and indicates that a user is likely to trust users with higher statuses than users with lower statuses. Modeling the status theory can potentially improve the prediction performance.

3.2.2 *Status Coefficient*

$\boldsymbol{R}$ is the status coefficient matrix among n users, defined as:

$$\boldsymbol{R} = \begin{pmatrix} \eta(1,1) & \eta(1,2) & \cdots & \eta(1,n) \\ \eta(2,1) & \eta(2,2) & \cdots & \eta(2,n) \\ \vdots & \vdots & \ddots & \vdots \\ \eta(n,1) & \eta(n,2) & \cdots & \eta(n,n) \end{pmatrix}.$$

Status coefficient $\eta(i,j)$ can be defined as:

$$\eta(i,j) = \begin{cases} s_i - s_j, \text{ if } j > i \text{ and } & Trust_{ij} > Trust_{ji}, \\ 0, \text{ otherwise}, \end{cases}$$

where $Trust_{ij} > Trust_{ji}$ means that u_i is more likely to establish a trust relation to u_j. s_i denotes the status score of u_i. In this work, we empirically find that the following definition of status difference between u_i and u_j works well for hsTrust:

$$s_i - s_j = \sqrt{\frac{1}{1+\log(r_i+1)} - \frac{1}{1+\log(r_j+1)}},$$

where $j > i$ and $s_i > s_j$. The function $s_i - s_j$ limits the values of status difference between u_i and u_j within [0,1]. Status score s_i plays an important role in the proposed framework, hsTrust, and we define $s_i \in [0,1]$. Status scores can be obtained by status ranking r_i, $r_i \in [1,n]$, and $r_i = 1$ denotes that u_i is the highest status ranking, $r_i = n$ denotes that u_i is the lowest status ranking.

There are many measurements for status ranking and in this subsection, we investigate the following three widely used status ranking measurements for social status coefficient.

• *Eigenvector Centrality* (*EC*). It assigns relative scores to all users in the network based on the concept that connections to high-scoring users contribute more to the score of the user in question than equal connections to low-scoring users. The centrality score x_i of user u_i can be formally defined:

$$x_i = \frac{1}{\lambda}\sum_{u_j \in M(u_i)} x_j = \frac{1}{\lambda}\sum_{u_j \in \boldsymbol{G}} \hat{\boldsymbol{G}}_{ij} x_j,$$

where $M(u_i)$ is a set of the neighbors of u_i, λ is a constant, and $\hat{\boldsymbol{G}}_{ij}$ is the transition matrix of the adjacency matrix $\boldsymbol{G}$ by normalizing each column to a sum of 1.

• *Number of Trustors* (*TON*). Trustors are the users who create the trust relations. This approach uses the number of trustors to measure the status ranking of users. The more the trustors, the higher the status ranking of a user. If the number of trustors for different users is the same, then we rank them randomly.

• *Number of Trustees* (*TEN*). Trustees are the users who manage the trust relations, usually appointed by the trustors. This approach uses the number of trustees to measure the status ranking of users. The more the trustees, the higher the status ranking of a user. If the number of trustees for different users is the same, then we rank them randomly.

4 Our Framework: hsTrust

In this section, we study how to model the homophily theory and the status theory in trust relations under the low-rank matrix tri-factorization model. After introducing homophily regularization and status regularization, we propose our framework with corresponding optimization method. Lastly, to verify the efficiency, we present the time complexity of our framework.

4.1 Low-Rank Matrix Tri-Factorization Model for Trust Prediction

Low-rank matrix tri-factorization seeks a "factor model" for representation using a low-rank approximation[18-20]. Mathematically, the matrix tri-factorization seeks a low-rank representation $\boldsymbol{U} \in \mathbb{R}^{n\times d}$ with $d \leqslant n$ for $\mathcal{U}$ via solving the following optimization problem:

$$\min_{\boldsymbol{U},\boldsymbol{H}} \ \|\boldsymbol{G} - \boldsymbol{U}\boldsymbol{H}\boldsymbol{U}^{\mathrm{T}}\|_{\mathrm{F}}^2,$$

where $\|.\|_{\mathrm{F}}^2$ is the Frobenius norm of a matrix. $\boldsymbol{U} \in \mathbb{R}^{n\times d}$ is the user preference matrix and d is the number of facets of user preferences. $\boldsymbol{H} \in \mathbb{R}^{d\times d}$ captures the more compact correlations among $\boldsymbol{U}$. $\boldsymbol{G}$ is approximated by three factors that specify soft membership of relations such as $G_{ij} = \boldsymbol{U}(i,:)\boldsymbol{H}\boldsymbol{U}^{\mathrm{T}}(j,:)$.

Regularization is one technique that is often used to control the over-fitting phenomenon, which involves adding a penalty term to the error function in order to discourage the coefficients from reaching large values. Low-rank matrix tri-factorization with respect to the Frobenius norm minimizes the sum squared differences to the target matrix. To avoid over-fitting, we add a smoothness regularization on $\boldsymbol{U}$ and $\boldsymbol{H}$, and then we have:

$$\min_{\boldsymbol{U},\boldsymbol{H}} \|\boldsymbol{G} - \boldsymbol{U}\boldsymbol{H}\boldsymbol{U}^{\mathrm{T}}\|_{\mathrm{F}}^2 + \alpha(\|\boldsymbol{U}\|_{\mathrm{F}}^2 + \|\boldsymbol{H}\|_{\mathrm{F}}^2),$$

where the term $(\|\boldsymbol{U}\|_{\mathrm{F}}^2 + \|\boldsymbol{H}\|_{\mathrm{F}}^2)$ is introduced to avoid over-fitting and the non-negative parameter α is used to control the capability of $\boldsymbol{U}$ and $\boldsymbol{H}$. Non-negative constraints are always applied to $\boldsymbol{U}$ and $\boldsymbol{H}$, then we have:

$$\begin{aligned}&\min_{\boldsymbol{U},\boldsymbol{H}} \|\boldsymbol{G} - \boldsymbol{U}\boldsymbol{H}\boldsymbol{U}^{\mathrm{T}}\|_{\mathrm{F}}^2 + \alpha(\|\boldsymbol{U}\|_{\mathrm{F}}^2 + \|\boldsymbol{H}\|_{\mathrm{F}}^2),\\ &s.t., \ \ \boldsymbol{U} \geqslant 0, \ \boldsymbol{H} \geqslant 0. \end{aligned} \tag{1}$$

It is easy to verify that (1) can model the properties of trust mentioned above and performance improvement is reported by [9,21-22] in terms of trust prediction. There are several nice properties of non-negative matrix factorization methods: 1) it results in intuitive meanings of the resultant matrices, and can be considered as a process of generating the original data by linear combinations of the latent features; 2) simple optimization methods such as gradient-based methods can be employed to find a well-worked optimal solution; 3) it has a nice probabilistic interpretation with Gaussian noise[23]; 4) it is very flexible and allows us to integrate prior knowledge such as homophily regularization and status regularization, introduced in Subsections 4.2 and 4.3.

4.2 Homophily Regularization Based on Low-Rank Matrix Tri-Factorization

The homophily theory supports that users with higher similarity are more likely to establish trust relations than those with lower similarity. We define $\zeta(i,j)$ as the homophily coefficient between u_i and u_j, satisfying: 1) $\zeta(i,j) \in [0,1]$; 2) $\zeta(i,j) = \zeta(j,i)$; 3) the larger $\zeta(i,j)$ is, the more likely a trust relation is established between u_i and u_j. With homophily coefficient, homophily regularization is to minimize the following term as:

$$\min \ \ \sum_{i=1}^{n}\sum_{j=1}^{n}\zeta(i,j)\|\boldsymbol{U}(i,:) - \boldsymbol{U}(j,:)\|_2^2.$$

Similar users in the low-rank space are more likely to establish trust relations[24] and their distances in the latent space are controlled by their homophily coefficients. For example, $\zeta(i,j)$ controls the latent distance between u_i and u_j. A larger value of $\zeta(i,j)$ indicates that u_i and u_j are more likely to establish trust relations according to the property 3) of homophily coefficient. Thus we force their latent representations as close as possible, while a smaller value of $\zeta(i,j)$ tells that the distance of their latent representations should be larger.

For a particular user u_i, the terms in homophily regularization related to his/her latent representation $\boldsymbol{U}(i,:)$ are:

$$\sum_{j=1}^{n}\zeta(i,j)\|\boldsymbol{U}(i,:) - \boldsymbol{U}(j,:)\|_2^2,$$

we can see that the latent representation for u_i is smoothed with other users, controlled by homophily coefficient; hence even for long tail users, with a few or even without any trust relations, we still can get an approximate estimate of their latent representations via homophily regularization, addressing the sparsity problem with traditional unsupervised methods.

After some derivations, we can get the matrix form of homophily regularization:

$$\begin{aligned}&\sum_{i=1}^{n}\sum_{j=1}^{n}\zeta(i,j)\|\boldsymbol{U}(i,:) - \boldsymbol{U}(j,:)\|_2^2\\ =&\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{d}\zeta(i,j)\big(\boldsymbol{U}(i,k) - \boldsymbol{U}(j,k)\big)^2\end{aligned}$$

$$= 2\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{d}\zeta(i,j)\boldsymbol{U}^2(i,k) -$$
$$2\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{d}\zeta(i,j)\boldsymbol{U}(i,k)\boldsymbol{U}(j,k)$$
$$= 2\sum_{k=1}^{d}\boldsymbol{U}^{\mathrm{T}}(:,k)(\boldsymbol{Q}-\boldsymbol{Z})\boldsymbol{U}(:,k)$$
$$= 2Trace(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{U}),$$

where $\boldsymbol{L} = \boldsymbol{Q} - \boldsymbol{Z}$ is the Laplacian matrix and $\boldsymbol{Q}$ is a diagonal matrix with the i-th diagonal element $Q(i,i) = \sum_{j=1}^{n} Z(j,i)$.

4.3 Status Regularization Based on Low-Rank Matrix Tri-Factorization

The status theory suggests that users with lower statuses are more likely to trust users with higher statuses. For a pair of users u_i and u_j, the likelihood of a trust relation established from u_i to u_j is calculated as $\boldsymbol{U}_i\boldsymbol{H}\boldsymbol{U}_j^{\mathrm{T}}$ under the framework. To model the status theory, we first perform algorithm to rank users from the perspective of social relations. We consider the following four cases for each pair u_i and u_j:

- *Case* 1: $s_i \geqslant s_j$ and $\boldsymbol{U}_i\boldsymbol{H}\boldsymbol{U}_j^{\mathrm{T}} > \boldsymbol{U}_j\boldsymbol{H}\boldsymbol{U}_i^{\mathrm{T}}$;
- *Case* 2: $s_i \geqslant s_j$ and $\boldsymbol{U}_i\boldsymbol{H}\boldsymbol{U}_j^{\mathrm{T}} \leqslant \boldsymbol{U}_j\boldsymbol{H}\boldsymbol{U}_i^{\mathrm{T}}$;
- *Case* 3: $s_i \leqslant s_j$ and $\boldsymbol{U}_i\boldsymbol{H}\boldsymbol{U}_j^{\mathrm{T}} \geqslant \boldsymbol{U}_j\boldsymbol{H}\boldsymbol{U}_i^{\mathrm{T}}$;
- *Case* 4: $s_i \leqslant s_j$ and $\boldsymbol{U}_i\boldsymbol{H}\boldsymbol{U}_j^{\mathrm{T}} < \boldsymbol{U}_j\boldsymbol{H}\boldsymbol{U}_i^{\mathrm{T}}$.

When $s_i \geqslant s_j$, the status theory suggests that the likelihood of a trust relation from u_j to u_i should be no smaller than that of a trust relation from u_i to u_j, i.e., $\boldsymbol{U}_i\boldsymbol{H}\boldsymbol{U}_j^{\mathrm{T}} \leqslant \boldsymbol{U}_j\boldsymbol{H}\boldsymbol{U}_i^{\mathrm{T}}$. Similarly when $s_i \leqslant s_j$, the likelihood of a trust relation from u_j to u_i should be no larger than that of a trust relation from u_i to u_j, i.e., $\boldsymbol{U}_i\boldsymbol{H}\boldsymbol{U}_j^{\mathrm{T}} \geqslant \boldsymbol{U}_j\boldsymbol{H}\boldsymbol{U}_i^{\mathrm{T}}$. Therefore among above four cases, case 2 and case 3 satisfy the status theory, while case 1 and case 4 contradict the status theory. Above analysis paves a way for us to model the status theory.

Based on case 2 and case 3, the status theory suggests that $(s_i - s_j)(\boldsymbol{U}_i\boldsymbol{H}\boldsymbol{U}_j - \boldsymbol{U}_j\boldsymbol{H}\boldsymbol{U}_i^{\mathrm{T}})$ should be no larger than 0. Therefore, we propose status regularization to model the status theory as:

$$\sum_{i}^{n}\sum_{j\neq i}(\max\{0,(s_i-s_j)(\boldsymbol{U}_i\boldsymbol{H}\boldsymbol{U}_j^{\mathrm{T}}-\boldsymbol{U}_j\boldsymbol{H}\boldsymbol{U}_i^{\mathrm{T}})\})^2. \quad (2)$$

Next we will show that by minimizing (2), we can model status theory as:

- case 2 and case 3 satisfy the status theory where $(s_i - s_j)(\boldsymbol{U}_i\boldsymbol{H}\boldsymbol{U}_j - \boldsymbol{U}_j\boldsymbol{H}\boldsymbol{U}_i^{\mathrm{T}}) \leqslant 0$. Therefore status regularization is 0, which means that we do not add any penalty on these cases;
- case 1 and case 4 contradict the status theory where $(s_i - s_j)(\boldsymbol{U}_i\boldsymbol{H}\boldsymbol{U}_j - \boldsymbol{U}_j\boldsymbol{H}\boldsymbol{U}_i^{\mathrm{T}}) > 0$. Then status regularization is $(s_i - s_j)(\boldsymbol{U}_i\boldsymbol{H}\boldsymbol{U}_j - \boldsymbol{U}_j\boldsymbol{H}\boldsymbol{U}_i^{\mathrm{T}})$, and minimizing this term will push $\boldsymbol{U}_i\boldsymbol{H}\boldsymbol{U}_j^{\mathrm{T}}$ close to $\boldsymbol{U}_j\boldsymbol{H}\boldsymbol{U}_i^{\mathrm{T}}$ and force the likelihood from a high status user to a low status user to be no larger than that from a low status user to a high status user, which can mitigate case 1 and case 4.

The above observations suggest that by minimizing status regularization, we can model the status theory. Since $(s_i - s_j)(\boldsymbol{U}_i\boldsymbol{H}\boldsymbol{U}_j^{\mathrm{T}} - \boldsymbol{U}_j\boldsymbol{H}\boldsymbol{U}_i^{\mathrm{T}})$ is equivalent to $(s_j - s_i)(\boldsymbol{U}_j\boldsymbol{H}\boldsymbol{U}_i^{\mathrm{T}} - \boldsymbol{U}_i\boldsymbol{H}\boldsymbol{U}_j^{\mathrm{T}})$, the status regularization can be rewritten as:

$$\sum_{i=1}^{n}\sum_{j=i+1}^{n}(\max\{0,(s_i-s_j)(\boldsymbol{U}_i\boldsymbol{H}\boldsymbol{U}_j^{\mathrm{T}}-\boldsymbol{U}_j\boldsymbol{H}\boldsymbol{U}_i^{\mathrm{T}})\})^2. \quad (3)$$

The optimization problem in (3) is jointly convex with respect to $\boldsymbol{U}$ and $\boldsymbol{H}$. As mentioned above, we assume that users in $\boldsymbol{U}$ are sorted by their status scores in descending order, if $i < j$, then $s_i > s_j$. Since $\boldsymbol{U}_i\boldsymbol{H}\boldsymbol{U}_j^{\mathrm{T}} = \boldsymbol{U}_j\boldsymbol{H}^{\mathrm{T}}\boldsymbol{U}_i^{\mathrm{T}}, \boldsymbol{U}_i\boldsymbol{H}\boldsymbol{U}_j^{\mathrm{T}} - \boldsymbol{U}_j\boldsymbol{H}\boldsymbol{U}_i^{\mathrm{T}}$ can be rewritten as $\boldsymbol{U}_j\boldsymbol{H}^{\mathrm{T}}\boldsymbol{U}_i^{\mathrm{T}} - \boldsymbol{U}_j\boldsymbol{H}\boldsymbol{U}_i^{\mathrm{T}}$. (3) can also be rewritten as:

$$\sum_{i=1}^{n}\sum_{j=i+1}^{n}(\max\{0,(s_i-s_j)(\boldsymbol{U}_i\boldsymbol{H}\boldsymbol{U}_j^{\mathrm{T}}-\boldsymbol{U}_j\boldsymbol{H}\boldsymbol{U}_i^{\mathrm{T}})\})^2$$
$$= \|\boldsymbol{R}\odot(\boldsymbol{U}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}-\boldsymbol{U}\boldsymbol{H}\boldsymbol{U}^{\mathrm{T}})\|_{\mathrm{F}}^2,$$

where $\odot$ is the Hadamard product where $(\boldsymbol{A}\odot\boldsymbol{B})_{ij} = A_{ij}\times B_{ij}$ for any two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ with the same size.

4.4 Combining Homophily Theory and Status Theory for Trust Prediction

With above solutions, we propose a novel framework, hsTrust, exploiting the homophily theory and the status theory simultaneously based on the low-rank matrix tri-factorization method, and the proposed framework is to solve the follow optimization problem:

$$\begin{aligned}&\min_{\boldsymbol{U},\boldsymbol{H}}\ \|\boldsymbol{W}\odot(\boldsymbol{G}-\boldsymbol{U}\boldsymbol{H}\boldsymbol{U}^{\mathrm{T}})\|_{\mathrm{F}}^2 + \\ &\alpha(\|\boldsymbol{U}\|_{\mathrm{F}}^2+\|\boldsymbol{H}\|_{\mathrm{F}}^2) + 2\lambda_1 Tr(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{U}) + \\ &\lambda_2\|\boldsymbol{R}\odot(\boldsymbol{U}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}-\boldsymbol{U}\boldsymbol{H}\boldsymbol{U}^{\mathrm{T}})\|_{\mathrm{F}}^2, \\ &\qquad s.t.,\ \ \boldsymbol{U}\geqslant 0,\ \boldsymbol{H}\geqslant 0,\end{aligned} \quad (4)$$

where the third term is homophily regularization to model the homophily theory, and the forth term is status regularization to model the status theory. The parameter λ_1 is introduced to control the contribution of homophily regularization, and λ_2 is introduced to control the contribution of status regularization in trust prediction. $\boldsymbol{W} \in \mathbb{R}^{n\times n}$ is constructed as:

$$W_{ij} = \begin{cases} 1, & \text{if } (\zeta(i,j) \cup G(i,j)) \neq 0, \\ c, & \text{if } (\zeta(i,j) \cup G(i,j)) = 0, \end{cases}$$

where c is a constant within 0 and 1.

We use $\mathcal{L}_k$ to denote the Lagrangian function of (4) in the k-th iteration, which can be written as:

$$\begin{aligned}\mathcal{L}_k = {} & \|\boldsymbol{W} \odot (\boldsymbol{G} - \boldsymbol{UHU}^{\mathrm{T}})\|_{\mathrm{F}}^2 + \alpha(\|\boldsymbol{U}\|_{\mathrm{F}}^2 + \|\boldsymbol{H}\|_{\mathrm{F}}^2) + \\ & 2\lambda_1 Tr(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{LU}) + \lambda_2\|\boldsymbol{R} \odot (\boldsymbol{UH}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}} - \\ & \boldsymbol{UHU}^{\mathrm{T}})\|_{\mathrm{F}}^2 - Tr(\varLambda^1\boldsymbol{U}) - Tr(\varLambda^2\boldsymbol{H}),\end{aligned}$$

where $\varLambda^1$ and $\varLambda^2$ are Lagrangian multipliers for non-negativity of $\boldsymbol{U}$ and $\boldsymbol{H}$, respectively.

By moving constants, $\mathcal{L}_k$ can be rewritten as:

$$\begin{aligned}\mathcal{L}_k = {} & -2Tr((\boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{G}^{\mathrm{T}})\boldsymbol{UHU}^{\mathrm{T}}) + \\ & Tr((\boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{UH}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{UHU}^{\mathrm{T}}) + \\ & \alpha Tr(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U} + \boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}) + 2\lambda_1 Tr(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{LU}) + \\ & 2\lambda_2 Tr(\boldsymbol{UH}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}(\boldsymbol{R} \odot \boldsymbol{R} \odot \boldsymbol{UH}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})) - \\ & 2\lambda_2 Tr(\boldsymbol{UH}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}(\boldsymbol{R} \odot \boldsymbol{R} \odot \boldsymbol{UHU}^{\mathrm{T}})) - \\ & Tr(\varLambda^1\boldsymbol{U}) - Tr(\varLambda^2\boldsymbol{H}).\end{aligned}$$

The KKT complementary condition[25] is:

$$\begin{aligned}& U_{ik}\varLambda^1_{ik} = 0, \\ & H_{ik}\varLambda^2_{ik} = 0, \\ & \forall\ i \in [1,n],\ k \in [1,d].\end{aligned} \tag{5}$$

Setting $\frac{\partial \mathcal{L}_k}{\partial \boldsymbol{U}} = 0$, $\frac{\partial \mathcal{L}_k}{\partial \boldsymbol{H}} = 0$, and using the KKT complementary condition in (5), we have:

$$\begin{aligned}& [(\boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{UH}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{UH} + \\ & (\boldsymbol{W} \odot \boldsymbol{W} \odot \boldsymbol{UHU}^{\mathrm{T}})\boldsymbol{UH}^{\mathrm{T}} + 2\alpha\boldsymbol{U} + \\ & (\boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{UHU}^{\mathrm{T}})\boldsymbol{UH}^{\mathrm{T}} + \\ & (\boldsymbol{W} \odot \boldsymbol{W} \odot \boldsymbol{UH}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{UH} + 4\lambda_1\boldsymbol{QU} + \\ & 4\lambda_2(\boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{UHU}^{\mathrm{T}})\boldsymbol{UH} + \\ & 4\lambda_2(\boldsymbol{R} \odot \boldsymbol{R} \odot \boldsymbol{UH}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{UH}^{\mathrm{T}} - \\ & 2(\boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{G}^{\mathrm{T}})\boldsymbol{UH} - \\ & 2(\boldsymbol{W} \odot \boldsymbol{W} \odot \boldsymbol{G})\boldsymbol{UH}^{\mathrm{T}} - 4\lambda_1\boldsymbol{ZU} - \\ & 2\lambda_2(\boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{UHU}^{\mathrm{T}})\boldsymbol{UH}^{\mathrm{T}} - \\ & 2\lambda_2(\boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{UH}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{UH} - \\ & 2\lambda_2(\boldsymbol{R} \odot \boldsymbol{R} \odot \boldsymbol{UH}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{UH} - \\ & 2\lambda_2(\boldsymbol{R} \odot \boldsymbol{R} \odot \boldsymbol{UHU}^{\mathrm{T}})\boldsymbol{UH}^{\mathrm{T}}]_{ik}U_{ik} = 0,\end{aligned}$$

$$\begin{aligned}& [\boldsymbol{U}^{\mathrm{T}}(\boldsymbol{W} \odot \boldsymbol{W} \odot \boldsymbol{UHU}^{\mathrm{T}})\boldsymbol{U} + \\ & \boldsymbol{U}^{\mathrm{T}}(\boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{UHU}^{\mathrm{T}})\boldsymbol{U} + \\ & 4\lambda_2\boldsymbol{U}^{\mathrm{T}}(\boldsymbol{R} \odot \boldsymbol{R} \odot \boldsymbol{UH}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U} + \\ & 2\alpha\boldsymbol{H} - 2\lambda_2\boldsymbol{U}^{\mathrm{T}}(\boldsymbol{R} \odot \boldsymbol{R} \odot \boldsymbol{UHU}^{\mathrm{T}})\boldsymbol{U} - \\ & 2\lambda_2\boldsymbol{U}^{\mathrm{T}}(\boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{UHU}^{\mathrm{T}})\boldsymbol{U} - \\ & 2\boldsymbol{U}^{\mathrm{T}}(\boldsymbol{W} \odot \boldsymbol{W} \odot \boldsymbol{G})\boldsymbol{U}]_{ik}H_{ik} = 0.\end{aligned}$$

In this work, we adopt an alternative optimization schema under which we update $\boldsymbol{U}$ and $\boldsymbol{H}$ with the following updating rules[26]:

$$U_{ik} \leftarrow U_{ik}\sqrt{\frac{A_{ik}}{B_{ik}}},\quad H_{ik} \leftarrow H_{ik}\sqrt{\frac{C_{ik}}{D_{ik}}},$$

where $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{D}$ are defined as:

$$\begin{aligned}\boldsymbol{A} = {} & 2(\boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{G}^{\mathrm{T}})\boldsymbol{UH} + \\ & 2(\boldsymbol{W} \odot \boldsymbol{W} \odot \boldsymbol{G})\boldsymbol{UH}^{\mathrm{T}} + 4\lambda_1\boldsymbol{ZU} + \\ & 2\lambda_2(\boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{UHU}^{\mathrm{T}})\boldsymbol{UH}^{\mathrm{T}} + \\ & 2\lambda_2(\boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{UH}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{UH} + \\ & 2\lambda_2(\boldsymbol{R} \odot \boldsymbol{R} \odot \boldsymbol{UH}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{UH} + \\ & 2\lambda_2(\boldsymbol{R} \odot \boldsymbol{R} \odot \boldsymbol{UHU}^{\mathrm{T}})\boldsymbol{UH}^{\mathrm{T}}, \\ \boldsymbol{B} = {} & (\boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{UH}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{UH} + \\ & (\boldsymbol{W} \odot \boldsymbol{W} \odot \boldsymbol{UHU}^{\mathrm{T}})\boldsymbol{UH}^{\mathrm{T}} + 2\alpha\boldsymbol{U} + \\ & (\boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{UHU}^{\mathrm{T}})\boldsymbol{UH}^{\mathrm{T}} + \\ & (\boldsymbol{W} \odot \boldsymbol{W} \odot \boldsymbol{UH}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{UH} + 4\lambda_1\boldsymbol{QU} + \\ & 4\lambda_2(\boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{UHU}^{\mathrm{T}})\boldsymbol{UH} + \\ & 4\lambda_2(\boldsymbol{R} \odot \boldsymbol{R} \odot \boldsymbol{UH}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{UH}^{\mathrm{T}}, \\ \boldsymbol{C} = {} & 2\lambda_2\boldsymbol{U}^{\mathrm{T}}(\boldsymbol{R} \odot \boldsymbol{R} \odot \boldsymbol{UHU}^{\mathrm{T}})\boldsymbol{U} + \\ & 2\lambda_2\boldsymbol{U}^{\mathrm{T}}(\boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{UHU}^{\mathrm{T}})\boldsymbol{U} + \\ & 2\boldsymbol{U}^{\mathrm{T}}(\boldsymbol{W} \odot \boldsymbol{W} \odot \boldsymbol{G})\boldsymbol{U}, \\ \boldsymbol{D} = {} & \boldsymbol{U}^{\mathrm{T}}(\boldsymbol{W} \odot \boldsymbol{W} \odot \boldsymbol{UHU}^{\mathrm{T}})\boldsymbol{U} + \\ & \boldsymbol{U}^{\mathrm{T}}(\boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{UHU}^{\mathrm{T}})\boldsymbol{U} + \\ & 4\lambda_2\boldsymbol{U}^{\mathrm{T}}(\boldsymbol{R} \odot \boldsymbol{R} \odot \boldsymbol{UH}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U} + 2\alpha\boldsymbol{H}.\end{aligned}$$

We can verify that the updating rules in (5) satisfy the above KKT condition. Since all matrices in (5) are non-negative, $\boldsymbol{U}$ and $\boldsymbol{H}$ are non-negative during the updating process. We also can prove that the updating rules in (5) are guaranteed to converge.

After learning $\boldsymbol{U}$ and $\boldsymbol{H}$, hsTrust suggests the likelihood of a trust relation established from u_i to u_j as $U_i\boldsymbol{HU}_j^{\mathrm{T}}$, namely, $\tilde{\boldsymbol{G}} = \boldsymbol{UHU}^{\mathrm{T}}$ is the new low-rank representation of $\boldsymbol{G}$.

4.5 Time Complexity

At each iteration, the major operations are to calculate partial deviations for $\boldsymbol{U}$ and $\boldsymbol{H}$. However, the high cost of the updating rules for $\boldsymbol{U}$ and $\boldsymbol{H}$ may limit the applications of the proposed algorithm, so it is essential to analyze the time complexity.

First we consider the time complexity of $\boldsymbol{A} = 2(\boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{G}^{\mathrm{T}})\boldsymbol{U}\boldsymbol{H} + 2(\boldsymbol{W} \odot \boldsymbol{W} \odot \boldsymbol{G})\boldsymbol{U}\boldsymbol{H}^{\mathrm{T}} + 4\lambda_1 \boldsymbol{Z}\boldsymbol{U} + 2\lambda_2(\boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{U}\boldsymbol{H}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U}\boldsymbol{H}^{\mathrm{T}} + 2\lambda_2(\boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{U}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U}\boldsymbol{H} + 2\lambda_2(\boldsymbol{R} \odot \boldsymbol{R} \odot \boldsymbol{U}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U}\boldsymbol{H} + 2\lambda_2(\boldsymbol{R} \odot \boldsymbol{R} \odot \boldsymbol{U}\boldsymbol{H}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U}\boldsymbol{H}^{\mathrm{T}}$. Obviously, terms like $(\boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{G}^{\mathrm{T}})\boldsymbol{U}\boldsymbol{H}$ and $(\boldsymbol{W} \odot \boldsymbol{W} \odot \boldsymbol{G})\boldsymbol{U}\boldsymbol{H}^{\mathrm{T}}$ have the same time complexity, and terms like $(\boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{U}\boldsymbol{H}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U}\boldsymbol{H}^{\mathrm{T}}$, $(\boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{U}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U}\boldsymbol{H}$, $(\boldsymbol{R} \odot \boldsymbol{R} \odot \boldsymbol{U}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U}\boldsymbol{H}$ and $(\boldsymbol{R} \odot \boldsymbol{R} \odot \boldsymbol{U}\boldsymbol{H}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U}\boldsymbol{H}^{\mathrm{T}}$ have the same time complexity; hence we can only compute the complexity of $(\boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{G}^{\mathrm{T}})\boldsymbol{U}\boldsymbol{H}$, $\boldsymbol{Z}\boldsymbol{U}$ and $(\boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{U}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U}\boldsymbol{H}$ in $\boldsymbol{A}$ separately. The matrix representation of trust relations $\boldsymbol{G}$ is very sparse, and thus term $(\boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{G}^{\mathrm{T}})\boldsymbol{U}\boldsymbol{H}$ can be computed with $O(nd^2)$. For $\lambda_1 \boldsymbol{Z}\boldsymbol{U}$, since we only consider the top similar users, we can obtain a sparse homophily coefficient matrix $\boldsymbol{Z}$, thus term $\boldsymbol{Z}\boldsymbol{U}$ can be computed with $O(nd^2)$. For $(\boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{U}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U}\boldsymbol{H}$, we can use $\boldsymbol{U}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{H}$ to replace $(\boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{U}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U}\boldsymbol{H}$ when computing time complexity. We can compute $\boldsymbol{U}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{H}$ by either:

$$(((\boldsymbol{U}\boldsymbol{H}^{\mathrm{T}})\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U})\boldsymbol{H}, \text{or}, \boldsymbol{U}(\boldsymbol{H}^{\mathrm{T}}((\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U})\boldsymbol{H})).$$

The former takes $O(n^2d)$ operations, while the latter costs $O(nd^2)$. As $d \ll n$, the latter is more efficient. Hence, we compute $\boldsymbol{U}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{H}$ as $\boldsymbol{U}(\boldsymbol{H}^{\mathrm{T}}((\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U})\boldsymbol{H}))$ with $O(nd^2)$ operations. The final time complexity of $\boldsymbol{A}$ is the the maximal complexity of all terms in $\boldsymbol{A}$, namely, $O(nd^2)$.

For $\boldsymbol{B} = (\boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{U}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U}\boldsymbol{H} + (\boldsymbol{W} \odot \boldsymbol{W} \odot \boldsymbol{U}\boldsymbol{H}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U}\boldsymbol{H}^{\mathrm{T}} + (\boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{U}\boldsymbol{H}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U}\boldsymbol{H}^{\mathrm{T}} + (\boldsymbol{W} \odot \boldsymbol{W} \odot \boldsymbol{U}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U}\boldsymbol{H} + 2\alpha\boldsymbol{U} + 4\lambda_1\boldsymbol{Q}\boldsymbol{U} + 4\lambda_2(\boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{U}\boldsymbol{H}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U}\boldsymbol{H} + 4\lambda_2(\boldsymbol{R} \odot \boldsymbol{R} \odot \boldsymbol{U}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U}\boldsymbol{H}^{\mathrm{T}}$. We can observe that terms except $\boldsymbol{U}$ and $\boldsymbol{Q}\boldsymbol{U}$ have the same time complexity. For $\boldsymbol{Q}\boldsymbol{U}$, $\boldsymbol{Q}$ is also sparse since $\boldsymbol{Z}$ is sparse. Term $\boldsymbol{U}$ can be computed with $O(nd)$. Term $(\boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{U}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U}\boldsymbol{H}$ can be computed with $O(nd^2)$; hence the final time complexity of $\boldsymbol{B}$ is the maximal complexity of all terms in $\boldsymbol{B}$, namely, $O(nd^2)$.

For $\boldsymbol{C} = 2\lambda_2\boldsymbol{U}^{\mathrm{T}}(\boldsymbol{R} \odot \boldsymbol{R} \odot \boldsymbol{U}\boldsymbol{H}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U} + 2\lambda_2\boldsymbol{U}^{\mathrm{T}}(\boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{R}^{\mathrm{T}} \odot \boldsymbol{U}\boldsymbol{H}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U} + 2\boldsymbol{U}^{\mathrm{T}}(\boldsymbol{W} \odot \boldsymbol{W} \odot \boldsymbol{G})\boldsymbol{U}$. The time complexity of $\boldsymbol{U}^{\mathrm{T}}(\boldsymbol{W} \odot \boldsymbol{W} \odot \boldsymbol{G})\boldsymbol{U}$ is $O(nd^2)$, and the time complexity of $\boldsymbol{U}^{\mathrm{T}}(\boldsymbol{R} \odot \boldsymbol{R} \odot \boldsymbol{U}\boldsymbol{H}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U}$ is namely to compute the complexity of $\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{H}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}$, which can be computed as $((\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U})\boldsymbol{H}(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}))$ with $O(nd^2)$ operations. The final time complexity of $\boldsymbol{C}$ is the maximal complexity of all terms in $\boldsymbol{C}$, namely, $O(nd^2)$.

For $\boldsymbol{D} = \boldsymbol{U}^{\mathrm{T}}(\boldsymbol{W} \odot \boldsymbol{W} \odot \boldsymbol{U}\boldsymbol{H}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U} + \boldsymbol{U}^{\mathrm{T}}(\boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{U}\boldsymbol{H}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U} + 4\lambda_2\boldsymbol{U}^{\mathrm{T}}(\boldsymbol{R} \odot \boldsymbol{R} \odot \boldsymbol{U}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U} + 2\alpha\boldsymbol{H}$. Terms like $\boldsymbol{U}^{\mathrm{T}}(\boldsymbol{W} \odot \boldsymbol{W} \odot \boldsymbol{U}\boldsymbol{H}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U}$, $\boldsymbol{U}^{\mathrm{T}}(\boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{W}^{\mathrm{T}} \odot \boldsymbol{U}\boldsymbol{H}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U}$ and $\boldsymbol{U}^{\mathrm{T}}(\boldsymbol{R} \odot \boldsymbol{R} \odot \boldsymbol{U}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})\boldsymbol{U}$ have the same time complexity, $O(nd^2)$, and the time complexity of $\boldsymbol{H}$ is $O(d^2)$. We select the maximal complexity of all terms in $\boldsymbol{D}$ as the complexity of $\boldsymbol{D}$, namely, $O(nd^2)$.

In summary, with above implementations, for each iteration, the time complexity is the maximal value from the time complexity of $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{D}$, namely, $O(nd^2)$. If m denotes iteration times, then the overall time complexity is $m \times O(nd^2)$.

In addition, we analyze the running time of hsTrust in different sizes of datasets, and the results are shown in Table 1.

Table 1. Running Time of hsTrust in Different Sizes in Epinions

Number of Users	Running Time (min)
1 000	36
2 000	40
3 000	56
4 000	65
5 000	75
6 000	86
7 000	98
7 936	112

We can see that it will consume more time with more users. The main reason is that more users will produce more trust relations, and the process of matrix decomposition will cost more time.

5 Experiments

In this section, we conduct experiments to evaluate the effectiveness of the proposed framework, hsTrust. Through the experiments, we aim to answer the following four questions:

- Do social theories improve the performance of trust prediction?
- How do homophily regularization and status regularization affect the proposed framework hsTrust?
- How do different measurements for homophily affect the proposed framework hsTrust?
- How do different measurements for social status affect the proposed framework hsTrust?

5.1 Datasets and Experiment Settings

5.1.1 Datasets

We use two real-world datasets from Epinions① and Ciao② for this study, which are general consumer review sites. Users can rate items by writing reviews and establish trust networks with their like-minded users.

We delete these users with less than three in-degrees and further filter the users with less than two reviews and ratings, aiming to obtain datasets that are large enough and have sufficient information for the purpose of evaluation. Some statistics of these two datasets are demonstrated in Table 2.

The distribution of trustees and trustors is demonstrated in Fig.3. Most users have few trustors and trustees, while a few users have an extremely high number of trustors and trustees, suggesting a power law distribution that is typical in social networks.

Also, Epinions and Ciao employ 5-star system to rate items and the rating distributions are shown in Fig.4(a) and Fig.4(b) for Epinions and Ciao, respectively. We note that the behavioral data is extremely skewed, and the majority of ratings are scores of 4 and 5 respectively, which demonstrate that users are likely to give positive ratings to items and reviews.

Table 2. Statistics of Epinions and Ciao

Dataset	Number of Users	Number of Trust Relations	Max. Number of Trustors	Max. Number of Trustees	Trust Network Density	Clustering Coefficient
Epinions	7 936	297 104	1 272	1 758	0.004 7	0.223 9
Ciao	5 635	107 492	100	785	0.003 4	0.225 2

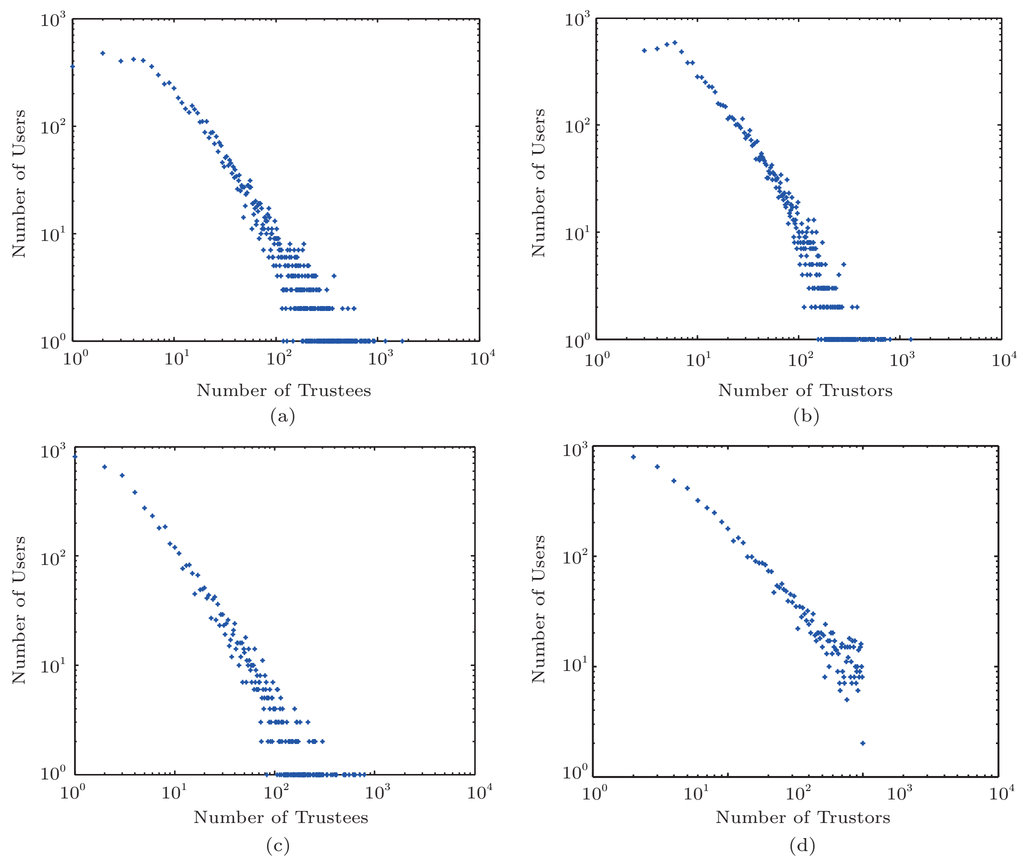


Fig.3. Distribution of trustees and trustors in Epinions and Ciao. (a) Trustees in Epinions. (b) Trustors in Epinions. (c) Trustees in Ciao. (d) Trustors in Ciao.

①http://www.epinions.com, May 2015.

②http://www.ciao.com, May 2015.

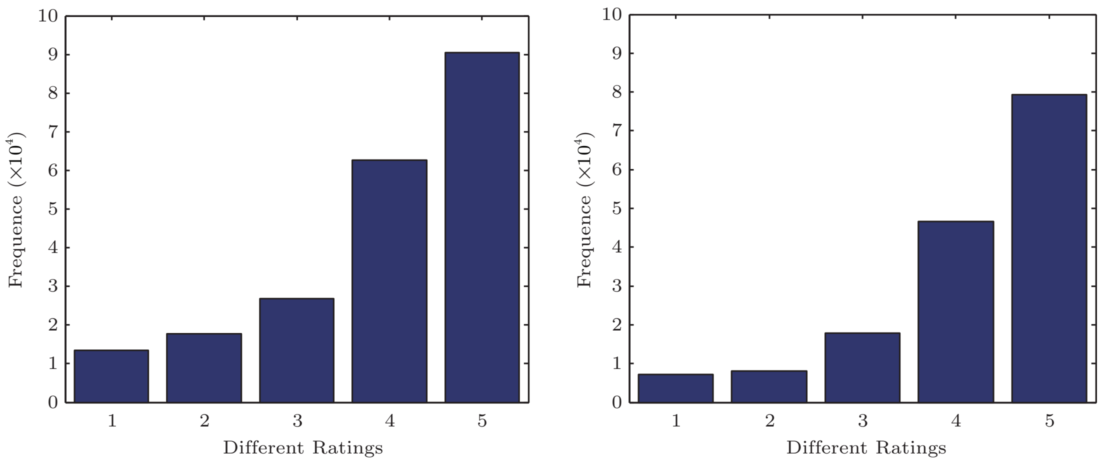


Fig.4. Distribution of different ratings in (a) Epinions and (b) Ciao.

5.1.2 *Experiment Settings*

We divide each dataset into two parts $\mathcal{A}$ and $\mathcal{B}$, where $\mathcal{A}$ is the set of user pairs with trust relations and $\mathcal{B}$ is the set of user pairs without trust relations. User pairs in $\mathcal{A}$ are sorted in chronological order in terms of the time when they established trust relations. We choose $x\%$ of $\mathcal{A}$ as old trust relations $\mathcal{C}$ and the remaining $1 - x\%$ as new trust relations $\mathcal{D}$ to predict. x is varied in $\{50, 60, 70, 80, 90\}$ in this paper and for each x, $|\mathcal{B}|=|\mathcal{D}|$. Table 3 shows the number of user pairs in $\mathcal{D}$ with different x.

Table 3. Number of User Pairs in $\mathcal{D}$ with Different x

x	User Pairs in Epinions	User Pairs in Ciao
50	139 552	53 746
60	111 641	42 997
70	83 731	32 248
80	55 821	21 494
90	27 910	10 749

We follow a common metric for unsupervised trust prediction in [27] to evaluate the performance of trust prediction. In detail, the trust predictor ranks user pairs in $\mathcal{D}\cup\mathcal{B}$ in decreasing order of confidence, and we take the first $|\mathcal{D}|$ pairs as the set of predicted trust relations, denoting as $\mathcal{E}$. Then, the trust prediction accuracy TP_{accuracy} can be calculated as:

$$TP_{\text{accuracy}} = \frac{|\mathcal{D}\cap\mathcal{E}|}{|\mathcal{D}|}.$$

We repeat the experiments five times and report the average performance.

5.2 Comparison of Different Trust Prediction Methods

To answer the first question, we compare hsTrust with various baseline methods:

• *TP:* trust relations are inferred through trust propagation by four types of atomic propagations: direct propagation, co-citation, transpose trust and trust coupling[28];

• *triNMF:* it performs a low-rank matrix tri-factorization on the user-user trust relation matrix without social status regularization and homophily regularization as shown in (1);

• *hTrust:* it performs a low-rank matrix tri-factorization on the user-user trust relation matrix with homophily regularization[9];

• *hsTrust:* it performs a low-rank matrix tri-factorization on the user-user trust relation matrix with homophily regularization and status regularization;

• *Random:* it generates trust relations without considering any rules, namely, it randomly suggests trust relations to pairs of users.

The parameters in above methods are determined via cross validation. For hsTrust, we choose rating similarity (RS) and eigenvector centrality (EC) to measure homophily coefficient and status coefficient, respectively. We empirically set $\alpha = 0.1$, $\lambda_1 = \lambda_2 = 0.1$ in Epinions and $\lambda_1 = \lambda_2 = 0.5$ in Ciao, $d = 200$, $c = 0.5$. More details about parameter analysis for hsTrust will be discussed in Subsections 5.3, 5.4 and 5.5. The comparison results are shown in Fig.5.

We have the following observations.

• Our proposed framework, hsTrust, performs consistently better than other baseline methods in Epinions

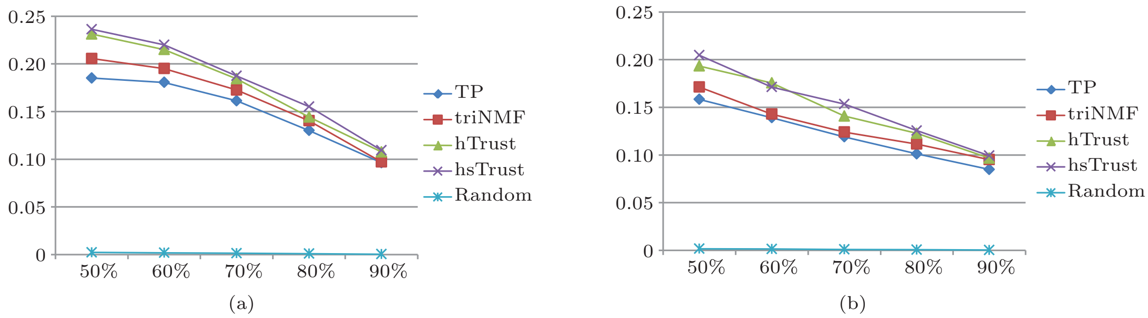


Fig.5. Performance comparisons in (a) Epinions and (b) Ciao.

and Ciao. It demonstrates the significance of our proposed framework by exploiting social theories.

• With the increase of x, the performance of all methods reduces. Generally, with more old trust relations, we should obtain better performance for the same set of new trust relations. However, in our experiments, since $\mathcal{B}$ is fixed, it becomes more difficult to predict $\mathcal{D}$ from $\mathcal{D}\cup\mathcal{B}$, and the new trust relations will be buried in a large amount of pairs without trust relations. Therefore, with the increase of x, the performance of all methods reduces.

• The performance of TP, triNMF, hTrust, and hsTrust is much better than that of Random, further demonstrating the existence of social theories in trust relations.

• Comparing triNMF, hTrust, and hsTrust with TP, we note that low-rank matrix tri-factorization methods can obtain better performance than trust propagation and improve the performance significantly.

• Our proposed framework, hsTrust, obtains better performance than triNMF, which demonstrates that homophily regularization and status regularization can improve the performance of trust prediction.

• Our proposed framework, hsTrust, obtains better performance than hTrust. As mentioned above, hTrust is a variant of the proposed framework without status regularization; hence, these results directly show that both of homophily regularization and status regularization can get better performance of trust prediction compared with only considering homophily regularization[9].

We perform t-test on all comparisons and the t-test results suggest that all improvement is significant. With these observations, we can answer the first question that the contributors to the performance improvement in hsTrust include: 1) considering the status theory; 2) considering the homophily theory; 3) incorporating low-rank matrix tri-factorization; hence, hsTrust is more effective than other baseline methods. More discussions about the effects of the homophily theory and the status theory on the proposed framework will be presented in the following subsections.

5.3 Impact of Social Theories

The parameters λ_1 and λ_2 are introduced to control the contribution from homophily regularization and status regularization for our proposed framework, hsTrust, respectively, and are needed to be further explored. Therefore, we investigate the impact of homophily regularization and status regularization via analyzing how the changes of λ_1 and λ_2 affect the performance of hsTrust in the terms of the trust prediction accuracy. Table 4 shows the performance of hsTrust with different λ_1 and λ_2.

Table 4. Trust Prediction Accuracy with Different λ_1 and λ_2

λ_1	λ_2	Epinions	Ciao
0.1	0.1	0.236	0.184
0.1	0.5	0.225	0.183
0.1	1.0	0.227	0.181
0.1	10.0	0.072	0.061
0.5	0.1	0.226	0.198
0.5	0.5	0.229	0.200
0.5	1.0	0.217	0.198
0.5	10.0	0.065	0.082
1.0	0.1	0.229	0.197
1.0	0.5	0.225	0.198
1.0	1.0	0.234	0.199
1.0	10.0	0.078	0.057
10.0	0.1	0.066	0.054
10.0	0.5	0.045	0.025
10.0	1.0	0.084	0.063
10.0	10.0	0.053	0.012

We draw the following observations.

• Intuitively, the performances of hsTrust at $\lambda_1 = \lambda_2$ are better than those of other parameter settings, especially, when $\lambda_1 = \lambda_2 = 0.1$ in Epinions and $\lambda_1 = \lambda_2 = 0.5$ in Ciao.

• When $\lambda_1 = 0$, it means to disable the impact of homophily effect. Similarly, when $\lambda_2 = 0$, it means to disable the impact of the status theory. The comparison results demonstrate that the performance of homophily effect is better than that of the status theory.

In our experiments, let $\lambda = \lambda_1 = \lambda_2$, which assumes that homophily regularization and status regularization play the same effect on hsTrust, and λ is varied in $\{0, 0.01, 0.1, 0.3, 0.5, 0.7, 1, 10\}$. The results are shown in Fig.6 for Epinions and Ciao, respectively.

We draw the following observations by comparing the results of different λ.

• With the increase of λ, the performance in both Epinions and Ciao shows similar patterns: first increases, reaches the peak value, and then degrades rapidly.

• When $\lambda = 0$, we eliminate the contribution from homophily regularization and status regularization on hsTrust, and the accuracy of trust prediction is lower than the peak performance. When λ is varied from 0 to 0.01, the accuracy of trust prediction changes much. The performance improves a lot with a little change on λ, which suggests that homophily regularization and status regularization can significantly improve the performance of trust prediction.

• When λ is very large, homophily regularization and status regularization dominate the learning process and the learned user preference matrix $\boldsymbol{U}$ and the characteristic correlations matrix $\boldsymbol{H}$ may be inaccurate.

The results suggest that the proposed framework, hsTrust, can achieve relatively good performance, and also further demonstrate the importance of modeling social theories in trust prediction, which correspondingly answer the second question presented in the beginning of Section 5. In summary, an appropriate combination of matrix tri-factorization and social theories can greatly improve the performance of trust prediction.

5.4 Impact of Different Measurements for Homophily

Homophily coefficient controls the distance of two users in the latent space. In this paper, we employ three widely used measures for homophily coefficient, i.e., Jaccard coefficient (JC), rating similarity (RS) and pearson correlation coefficient (PCC) in the proposed framework, hsTrust.

Table 5 demonstrates the performance of the proposed framework with different measures of homophily coefficient. In the table, Random means we randomly assign homophily coefficients within [0, 1].

From Table 5, we observe answers to the third question proposed at the beginning of Section 5.

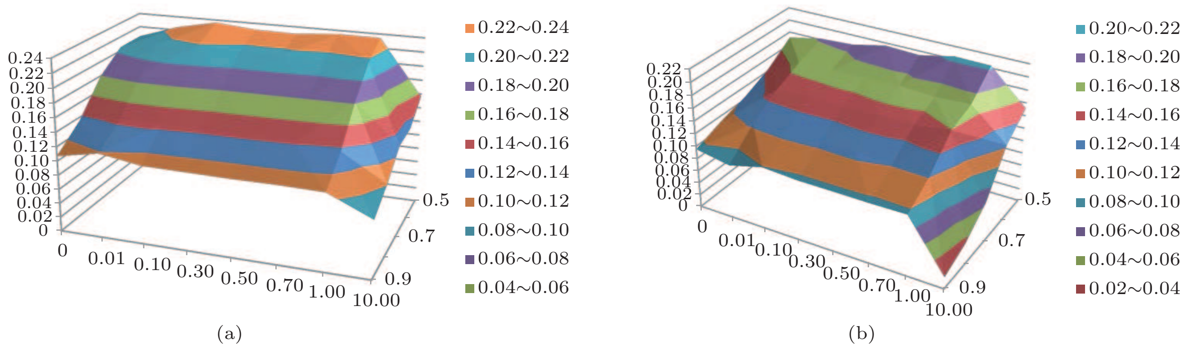


Fig.6. Performance comparisons in (a) Epinions and (b) Ciao.

Table 5. Different Measurements of Homophily

x (%)	Epinions				Ciao			
	JC	RS	PCC	Random	JC	RS	PCC	Random
50	0.228 0	0.236 2	0.229 8	0.192 5	0.198 7	0.204 6	0.199 8	0.169 8
60	0.206 2	0.219 9	0.208 9	0.172 4	0.163 5	0.171 2	0.168 9	0.151 4
70	0.176 4	0.187 4	0.179 5	0.153 6	0.142 4	0.159 7	0.151 2	0.121 1
80	0.139 0	0.145 2	0.141 0	0.117 0	0.118 9	0.125 6	0.123 4	0.094 7
90	0.107 2	0.118 5	0.115 2	0.082 1	0.090 6	0.099 3	0.095 0	0.074 5

• RS obtains the best performance among the three measures of homophily coefficient, i.e., JC, RS and PCC. We also note that RS and PCC always obtain better performance than JC, which suggests that different users might have different tastes to the same item.

• Random obtains the worst performance, suggesting that homophily coefficient should not be a random value.

5.5 Impact of Different Measurements for Social Status

Status defines how prestigious an individual is ranked in a social community. In this paper, we employ three widely used measures for social statuses, i.e., eigenvector centrality (EC), number of trustors (TON) and number of trustees (TEN) in the proposed framework, hsTrust.

Table 6 demonstrates the performance of hsTrust with different status measurements in Epinions and Ciao. In the table, Random means that we generate user ranking without considering any rules, namely, Random method randomly suggests user ranking.

From Table 6, we observe answers to the fourth question proposed at the beginning of Section 5.

• hsTrust with random status ranking always obtains the worst performance, suggesting that status ranking should not be a random value.

• Different measurements may lead to different performance for hsTrust. The performance of EC and TON is better than that of TEN.

6 Literature Review

In recent years, the notion of trust has attracted more and more attention from computer science communities, and many trust models are studied, which can mainly be classified as statistics analysis based approaches and machine learning based approaches.

Statistics analysis based approaches merge multiple dimensions, such as history, recommendation, context, reputation and so on, to achieve more accurate trustworthiness. In [29], the authors elaborated Unified Trust Model (UTM) which calculates entities' trustworthiness based on history, recommendation, context and platform integrity measurement, and formally uses these factors in trustworthiness calculation. Experiments demonstrated that UTM offered responsive behavior and could be used effectively in the low interaction environments. In [30], the authors proposed a trust model by investigating the four main antecedent influences on consumer trust in Internet shopping, a major form of business-to-consumer e-commerce: trustworthiness of the Internet merchant, trustworthiness of the Internet as a shopping medium, infrastructural (contextual) factors (e.g., security, third-party certification), and other factors (e.g., company size, demographic variables). In [31], the authors proposed a framework CSTrust for conducting cloud service trustworthiness evaluation by combining QoS prediction and customer satisfaction estimation. In [32], the authors proposed a trust model, which considers two aspects of trust: popularity trust (PopTrust) and engagement trust (EngTrust). Popularity trust[33] refers to the acceptance and approval of a member by others in the community, while engagement trust captures the involvement of someone in the community.

Machine learning based approaches construct trust model based on supervised learning and unsupervised learning. Supervised learning approaches regard modeling trust as a classification problem. Based on interaction data, a supervised method was proposed in [34] to distinguish strong ties from weak ties by predicting binary relationship strength between users. In [35], the authors proposed a time-aware trust prediction approach which incorporates the temporal evolution of trust networks to predict future trust relations (or links) with a supervised learning method. In [36], the authors proposed a classification approach to predict if a user trusts another user using features derived from his/her interactions with the latter as well as from the interactions with other users. In [37], the authors proposed a Bayesian model for event-based trust

Table 6. Different Measurements of Social Statuses

x (%)	Epinions				Ciao			
	JC	RS	PCC	Random	JC	RS	PCC	Random
50	0.2362	0.2342	0.2094	0.1954	0.2046	0.1911	0.1890	0.1835
60	0.2199	0.2160	0.1864	0.1698	0.1712	0.1625	0.1621	0.1536
70	0.1874	0.1853	0.1587	0.1468	0.1597	0.1389	0.1351	0.1118
80	0.1452	0.1450	0.1268	0.1157	0.1256	0.1230	0.1112	0.0856
90	0.1185	0.0983	0.0912	0.0798	0.0993	0.1001	0.0958	0.0751

and defined a mathematical measure for quantitatively comparing the effectiveness of probabilistic computational trust systems in various environments. In [38], the authors argued that trust is multi-faceted, and Bayesian networks provide a flexible method to present differentiated trust and combined different aspects of trust; hence, they proposed a Bayesian network based trust model in peer-to-peer networks. In the supervised model, the explicit trust value in a web of trust is necessary and critical to train the trust prediction model as an output variable; however, trust relations follow a power law distribution, making the classification problem extremely unbalanced[39] and thus affecting the accuracy of classification. Unsupervised learning approaches regard modeling trust as a propagation problem. In [40], the authors introduced an extended trust model for detecting malicious activities in online social networks. The major insight is to conduct a trust propagation process over a novel heterogeneous social graph which is able to model different social activities. In [41], the authors proposed a method to accurately predict trust relationships of a target user even if he/she did not have much interaction information, which considers positive, implicit, and negative information of all users in a network based on belief propagation to predict trust relationships of a target user. In [42], the authors investigated the strength of social influence in trust networks, which showed that the strength of trust relation correlates with the similarity among the users. A modified matrix factorization technique was used to estimate strengths of trust relations. In [28], the authors developed a formal framework of trust propagation schemes, which first separates trust and distrust matrix and then performs operations on them to obtain the transitive trust between two nodes. In [9], the authors proposed an unsupervised framework incorporating low-rank matrix factorization and homophily regularization for trust prediction, and the experimental results demonstrated the effectiveness of the proposed framework. In [43], the authors proposed a method to model and compute the bias or the truthfulness of a user in trust networks. The biases of users are their propensity to trust/distrust other users. They claimed that their model conformed well to other graph ranking algorithms and social theories such as the balance theory.

Although many theoretical models and systems have been developed for trust prediction, very few of them exploit social theories from the perspective of social science, and mainly rely on web of trust which is often too sparse in social media to predict trust relations with high accuracy.

7 Conclusions

Trust in a person is a commitment to an action based on a belief that the future actions of that person will lead to a good outcome; hence, we relied on trust to handle various threats brought by uncertainty and information overload in social media. In this paper, we studied the problem of trust prediction by exploiting social theories to infer unknown trust relations. A fine-grained framework, hsTrust, was proposed to capture trust relations in social media. Experimental results on two real-world datasets from Epinions and Ciao demonstrated that our proposed framework, hsTrust, outperforms the state-of-the-art baseline methods, and further experiments were conducted to understand the importance of the status theory and the homophily theory in the proposed framework.

There are several directions needing further investigation. Firstly, hsTrust does not consider temporal information related to trust networks and product ratings. Trust relations are likely to change over time and it is interesting to exploit temporal effects in hsTrust. Secondly, we plan to explore new models and algorithms on trust, and incorporate our models and algorithms to other applications.

References

[1] Guo G, Zhang J, Yorke-Smith N. Leveraging multiviews of trust and similarity to enhance clustering-based recommender system. *Knowledge-Based System*, 2015, 74: 14-27.

[2] Zhao T, Li C, Li M, Ding Q, Li L. Social recommendation incorporating topic mining and social trust analysis. In *Proc. the 22nd ACM CIKM*, Oct. 27-Nov. 1, 2013, pp.1643-1648.

[3] Lu Y, Tsaparas P, Ntoulas A, Polanyi L. Exploiting social context for review quality prediction. In *Proc. the 19th ACM WWW*, Apr. 2010, pp.691-700.

[4] Oh H J, Yoon Y C, Kim H K. Finding more trustworthy answers: Various trustworthiness factors in question answering. *Journal of Information Science*, 2013, 39(4): 509-522.

[5] Richardson M, Domingos P. Mining knowledge-sharing sites for viral marketing. In *Proc. the 8th ACM SIGKDD*, Jul. 2002, pp.61-70.

[6] Liu G, Wang Y. Trust-oriented service provider selection in complex online social networks. In *Advanced Web Services*, Bouguettaya A, Sheng Q Z, Daniel F (eds.), Springer New York, 2014, pp.363-380.

[7] Sherchan W, Nepal S, Paris C. A survey of trust in social networks. *ACM Computing Surveys* (*CSUR*), 2013, 45(4): Article No. 47.

[8] Xu T, Gao J, Xiao L, Regan A. Online classification using a voted RDA method. In *Proc. the 28th AAAI*, Jul. 2014, pp.2170-2176.

[9] Tang J, Gao H, Hu X, Liu H. Exploiting homophily effect for trust prediction. In *Proc. the 6th ACM WSDM*, Feb. 2013, pp.53-62.

[10] Yang S H, Smola A J, Long B, Zha H, Chang Y. Friend or frenemy? Predicting signed ties in social networks. In *Proc. the 35th ACM SIGIR*, Aug. 2012, pp.555-564.

[11] Tang W, Zhuang H, Tang J. Learning to infer social ties in large networks. In *Proc. ECML PKDD*, Sept. 2011, pp.381-397.

[12] Tang J, Lou T, Kleinberg J. Inferring social ties across heterogenous networks. In *Proc. the 5th ACM WSDM*, Feb. 2012, pp.743-752.

[13] Leskovec J, Huttenlocher D, Kleinberg J. Signed networks in social media. In *Proc. ACM CHI*, Apr. 2010, pp.1361-1370.

[14] McPherson M, Smith-Lovin L, Cook J M. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 2001, 27: 415-444.

[15] Ma H, Zhou D, Liu C, Lyu M, King I. Recommender systems with social regularization. In *Proc. the 4th ACM WSDM*, Feb. 2011, pp.287-296.

[16] Giddens A, Duneier M, Appelbaum R P, Carr D. Introduction to Sociology (8th edition). W.W. Norton & Company, 2012.

[17] Leskovec J. How status and reputation shape human evaluations: Consequences for recommender systems. In *Proc. the 22nd ACM WWW*, May 2013, pp.937-938.

[18] Lee D D, Seung H S. Learning the parts of objects by non-negative matrix factorization. *Nature*, 1999, 401(6755): 788-791.

[19] Lee D D, Seung H S. Algorithms for nonnegative matrix factorization. In *Proc. the NIPS*, Nov. 27-Dec. 2, 2000, pp.556-562.

[20] Ding C, Li T, Peng W, Park H. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proc. ACM KDD*, Aug. 2006, pp.126-135.

[21] Huang J, Nie F, Huang H, Lei Y, Ding C. Social trust prediction using rank-k matrix recovery. In *Proc. the 23rd AAAI*, Aug. 2013, pp.2647-2653.

[22] Huang J, Nie F, Huang H, Tu Y C. Trust prediction via aggregating heterogeneous social networks. In *Proc. the 21st ACM CIKM*, Oct. 29-Nov. 2, 2012, pp.1774-1778.

[23] Salakhutdinov R, Mnih A. Probabilistic matrix factorization. In *Proc. the 21st NIPS*, Dec. 2007, pp.1257-1264.

[24] Oyama S, Hayashi K, Kashima H. Cross-temporal link prediction. In *Proc. the 11th IEEE ICDM*, Dec. 2011, pp.1188-1193.

[25] Ye J J. Constraint qualifications and KKT conditions for bilevel programming problems. *Mathematics of Operations Research*, 2006, 31(4): 811-824.

[26] Ding C, Li T, Jordan M I. Nonnegative matrix factorization for combinatorial optimization: Spectral clustering, graph matching, and clique finding. In *Proc. the 8th IEEE ICDM*, Dec. 2008, pp.183-192.

[27] Liben-Nowell D, Kleinberg J. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 2007, 58(7): 1019-1031.

[28] Guha R, Kumar R, Raghavan P, Tomkins A. Propagation of trust and distrust. In *Proc. the 13th ACM WWW*, May 2004, pp.403-412.

[29] Khiabani H, Idris N B, Manan J l. A unified trust model for pervasive environments-simulation and analysis. *KSII Transactions on Internet and Information Systems*, 2013, 7(7): 1569-1584.

[30] Lee M K, Turban E. A trust model for consumer Internet shopping. *International Journal of Electronic Commerce*, 2001, 6: 75-92.

[31] Ding S, Yang S, Zhang Y, Liang C, Xia C. Combining QoS prediction and customer satisfaction estimation to solve cloud service trustworthiness evaluation problems. *Knowledge-Based Systems*, 2014, 56: 216-225.

[32] Nepal S, Paris C, Bista S, Sherchan W. A trust model-based analysis of social networks. *International Journal of Trust Management in Computing and Communications*, 2013, 1(1): 3-22.

[33] Caverlee J, Liu L, Webb S. The social trust framework for trusted social information management: Architecture and algorithms. *Information Sciences*, 2010, 180(1): 95-112.

[34] Kahanda I, Neville J. Using transactional information to predict link strength in online social networks. In *Proc. the 3rd ICWSM*, May 2009, pp.74-81.

[35] Zolfaghar K, Aghaie A. Evolution of trust networks in social web applications using supervised learning. *Procedia Computer Science*, 2011, 3: 833-839.

[36] Liu H, Lim E, Lauw H, Le M, Sun A, Srivastava J, Kim Y. Predicting trusts among users of online communities: An Epinions case study. In *Proc. the 9th ACM Conference on Electronic Commerce*, July 2008, pp.310-319.

[37] Nielsen M, Krukow K, Sassone V. A Bayesian model for event-based trust. *Electronic Notes in Theoretical Computer Science*, 2007, 172: 499-521.

[38] Wang Y, Vassileva J. Trust and reputation model in peer-to-peer networks. In *Proc. the 3rd Int. Conf. Peer-to-Peer Computing*, Sept. 2003, pp.150-157.

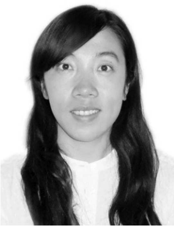
[39] Wang D, Pedreschi D, Song C, Giannotti F, Barabasi A L. Human mobility, social ties, and link prediction. In *Proc. the 17th ACM KDD*, Aug. 2011, pp.1100-1108.

[40] Agarwal M, Zhou B. Using trust model for detecting malicious activities in Twitter. In *Proc. the 7th Int. Conf. Social Computing, Behavioral-Cultural Modeling and Prediction*, Apr. 2014, pp.207-214.

[41] Jang M H, Faloutsos C, Kim S W. Trust prediction using positive, implicit, and negative information. In *Proc. the 23rd ACM WWW Companion*, Apr. 2014, pp.303-304.

[42] Au Yeung C, Iwata T. Strength of social influence in trust networks in product review sites. In *Proc. the 4th ACM WSDM*, Feb. 2011, pp.495-504.

[43] Mishra A, Bhattacharya A. Finding the bias and prestige of nodes in networks based on trust scores. In *Proc. the 20th ACM WWW*, Mar. 28-Apr. 1, 2011, pp.567-576.



Ying Wang is a lecturer of the College of Computer Science and Technology at Jilin University, Changchun. She obtained her Ph.D. degree in computer science at Jilin University in 2010. Her research interests are in data mining, machine learning, social computing and search engine.

Xin Wang got his Ph.D. degree in computer science from Jilin University, Changchun, in 2015. He received his Bachelor's degree in computer science from Harbin Engineering University, Harbin, in 2008. His research interests are in data mining, machine learning and social computing.

Wan-Li Zuo is a professor in Jilin University, Changchun. He received his B.S., M.S., and Ph.D. degrees in computer science from the same University. His research interests include data mining, information retrieval, natural language processing, machine learning, etc.