Exploiting Homophily Effect for Trust Prediction

Jiliang Tang, Huiji Gao, Xia Hu and Huan Liu
Computer Science and Engineering
Arizona State University
Tempe, AZ 85281
{Jiliang.Tang, Huiji.Gao, Xia.Hu, Huan.Liu}@asu.edu

ABSTRACT

Trust plays a crucial role for online users who seek reliable information. However, in reality, user-specified trust relations are very sparse, i.e., a tiny number of pairs of users with trust relations are buried in a disproportionately large number of pairs without trust relations, making trust prediction a daunting task. As an important social concept, however, trust has received growing attention and interest. Social theories are developed for understanding trust. Homophily is one of the most important theories that explain why trust relations are established. Exploiting the homophily effect for trust prediction provides challenges and opportunities. In this paper, we embark on the challenges to investigate the trust prediction problem with the homophily effect. First, we delineate how it differs from existing approaches to trust prediction in an unsupervised setting. Next, we formulate the new trust prediction problem into an optimization problem integrated with homophily, empirically evaluate our approach on two datasets from real-world product review sites, and compare with representative algorithms to gain a deep understanding of the role of homophily in trust prediction.

Categories and Subject Descriptors

H3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering*

General Terms

Algorithms, Design, Experimentation

Keywords

Trust Prediction, Homophily Effect, Homophily Regularization, Trust Networks, Social Correlation

1. INTRODUCTION

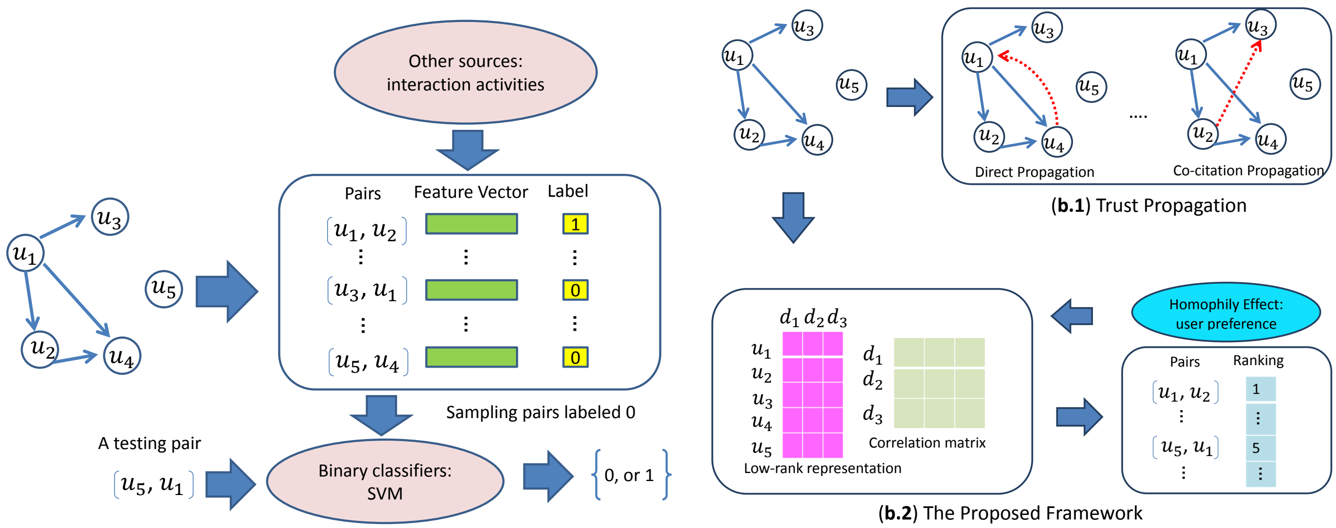
With the pervasive of social media, the explosion of user generated data makes the information overload problem increasingly severe. Trust, which provides information about from whom we should accept information and with whom we should share information [6], plays an important role in helping online users collect reliable information. For example, users in e-commerce are likely to gather information from their trusted users to make decisions. Hence, the trust mechanism is widely implemented by online service providers, especially e-commerce websites such as eBay[1] and product review websites like Epinions[2].

Recent years witness many trust related online applications, such as trust-aware recommendation systems [6, 12, 21], finding high-quality user generated content [11, 3] and viral marketing [19]. However, in reality, the available explicit trust relations are extremely sparse, and, online trust relations follow a power law distribution, suggesting that a small number of users specify many trust relations while a large proportion of users specify a few trust relations. Trust prediction is proposed to infer unknown trust relations, and it is important to address the problem of sparseness in user-specified trust relations.

Literature on trust prediction is rapidly growing [7, 10, 1, 17], roughly divided into two groups: unsupervised methods [7, 1] and supervised methods [10, 17]. As illustrated in Figure 1(a), after extracting features from available sources and considering the existence of trust as labels, supervised methods train a binary classifier. However, these methods have inherent limitations. The huge disproportion of pairs of users with (positive samples, labelled as 1 in Figure 1(a)) and without relations (negative samples, labelled as 0 in Figure 1(a)) makes the classification problem extremely imbalanced and the performance of these methods is sensitive to the sampled negative samples. While unsupervised methods such as trust propagation [7] are able to infer trust relations for two indirectly connected users as demonstrated in Figure 1(b.1). However, the power law distribution indicates that the available trust relations may not be enough to guarantee the success of these methods (u_5 in Figure 1(b.1)) [13, 22]. As a social concept, trust has received growing attention and there are many social theories are developed. Homophily is one of the most important theories that attempt to explain why people establish trust relations with each other [10]. 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 that effect provides a new perspective for trust prediction and enables us to do advanced research on trust prediction.

WSDM'13, February 4–8, 2013, Rome, Italy.

[1] http://www.ebay.com/
[2] http://www.epinions.com



(a) Supervised Trust Prediction (b) Unsupervised Trust Prediction

Figure 1: Supervised and Unsupervised Trust Prediction

In this paper, we study the problem of trust prediction by exploiting homophily effect. In essence, we investigate: (1) how to capture homophily effect in trust relations; and (2) how to take advantage of that effect for trust prediction. Our solutions to these two challenges result in a new framework, hTrust, for trust prediction. As demonstrated in Figure 1(b.2), hTrust does trust prediction in an unsupervised scenario by seeking low-rank representations for users and their correlations while exploiting homophily effect. Our main contributions are summarized next.

- Demonstrate the existence of homophily in trust relations: similar users are more likely to establish trust relations while trusted users are more similar;
- Provide an approach to exploit homophily effect in trust relations via homophily regularization;
- Propose an unsupervised framework, hTrust, for the problem of trust prediction, incorporating low-rank matrix factorization with homophily regularization; and
- Evaluate hTrust extensively using datasets from product review sites to understand the working of hTrust.

The rest of paper is organized as follows. Section 2 describes the datasets and verifies homophily in trust relations. Section 3 introduces how to employ the low-rank matrix factorization method for trust prediction. Section 4 introduces the details on homophily regularization and the proposed framework. Section 5 presents experimental results and our observations. Section 6 briefly reviews related work. Section 7 concludes this study with future work.

2. DATA ANALYSIS

We collect two publicly available datasets for this study, i.e., Epinions and Ciao[3]. Both sites are product review sites

Table 1: Statistics of the Datasets

	Epinions	Ciao
# of Users	8,527	6,262
# of items	26,552	20,416
# of Ratings	225,579	167,320
# of Trust Relations	302,177	109,524
Max # of Trustors	1,285	100
Max # of Trustees	1,805	797
Trust Network Density	0.0042	0.0028
Clustering Coefficient	0.2242	0.2254

where users can rate items by writing reviews and establish trust networks with their like-minded users.

We filter the users with less than two trustors and items with less than two ratings, aiming to obtain datasets that are large enough and have sufficient historical information for the purpose of evaluation. Some statistics of the datasets are shown in Table 1. On average, users of Epinions have 35.43 trust relations and 26.45 ratings, while users of Ciao have 17.49 trust relations and 26.72 ratings.

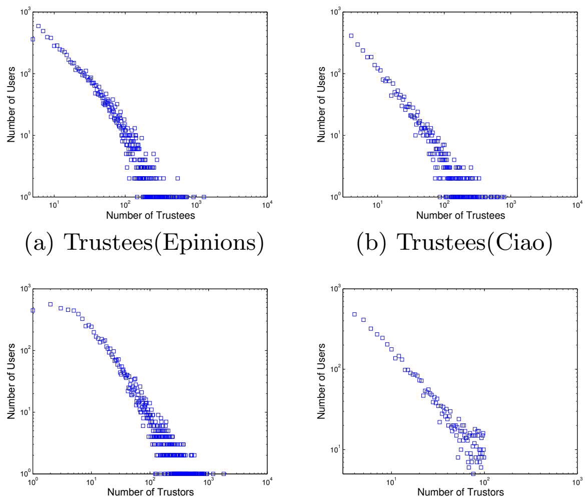
The distributions of trustors and trustees for each user are demonstrated in Figure 2. Most users have few trustors and trustees, while a few users have an extremely high number of trustors or trustees, suggesting a power law distribution that is typical in social networks.

Both Epinions and Ciao employ 5-star system to rate items and the rating distributions are shown in Figure 3(a) and Figure 3(b) for Epinions and Ciao, respectively. We note that the majority of ratings are scores of 4 and 5 and this observation is consistent with previous studies: users are likely to give positive ratings to items [21].

2.1 Homophily in Trust Relations

Homophily is one of the most important social correlation theories, observed in many social networks such as following relations in Twitter [24]. In this subsection, we investigate

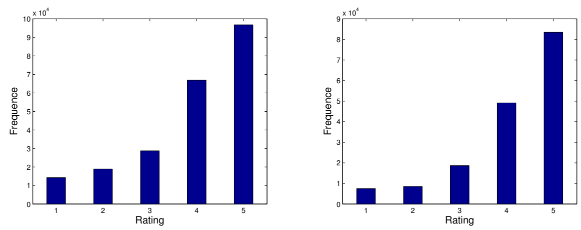
[3]These datasets are available from the first author webpage: http:// www.public.asu.edu/~jtang20/ datasetcode/truststudy.htm



(a) Trustees(Epinions) (b) Trustees(Ciao)

(c) Trustors(Epinions) (d) Trustors(Ciao)

Figure 2: Trustors and Trustees Distributions in Epinions and Ciao



(a) Epinions (b) Ciao

Figure 3: Rating Distributions in Epinions and Ciao

homophily in trust relations via studying the correlation between trust relations and users' similarity. Specifically, in the context of product review sites, we ask two questions:

- Are users with trust relations more similar in terms of their ratings than those without?
- Are users with higher similarity more likely to establish trust relations than those with lower similarity?

To answer the **first question**, we have to define users' rating similarity. In this work, we use the cosine similarity of users' rating vectors to measure their rating similarity. With this definition, we calculate two similarities for each trust relation, i.e., trust similarity ts and random similarity rs. For example, for the trust relation $u_i \rightarrow u_j$, indicating that u_i trusts u_j, ts is the rating similarity between u_i and u_j while rs is the similarity between u_i and a randomly chosen user without trust relations. Finally we obtain two similarity vectors, $\mathbf{s}_t$ and $\mathbf{s}_r$. $\mathbf{s}_t$ is the set of all trust similarities ts while $\mathbf{s}_r$ is the set of rs.

We conduct a two-sample t-test on $\mathbf{s}_t$ and $\mathbf{s}_r$. The null hypothesis is $H_0 : \mathbf{s}_t = \mathbf{s}_r$, and the alternative hypothesis is $H_1 : \mathbf{s}_t > \mathbf{s}_r$. For both datasets, the null hypothesis is rejected at significance level $\alpha = 0.01$ with p-value of $5.12e{-18}$ and $3.76e{-21}$ in Epinions and Ciao, respectively. The evidence from t-test suggests a positive answer to the first question: *with high probability, users with trust relations have higher rating similarity than those without.*

For the **second question**, we want to investigate if users with higher similarity at time t is more likely to establish trust relations at time $t+1$ than those with lower similarity. We study this problem in Epinions since it provides temporal information when ratings are created and when trust relations are established. The earliest rating was published on Jul 05, 1999 and the latest one was on May 08, 2011. However, temporal information about the trust relations established before Jan 11, 2001 is not available from Epinions. Therefore, we split the whole dataset into 11 timestamps, i.e., $\mathbf{t} = \{t_1, \ldots, t_{11}\}$, where t_1 covers the data before Jan 11, 2001, t_{11} contains data after Jan 11, 2010 and for t_2 to t_{10}, each of them contains data for one year.

For timestamp $t_i (1 \leq i \leq 10)$, we rank all pairs of users without trust relations in terms of rating similarity in descending order and then we pick out $x\%$ pairs from the top and $x\%$ pairs from the bottom to form a higher-similarity group, denoted as $\mathcal{G}_h^i$ and a lower-similarity group, represented by $\mathcal{G}_l^i$, respectively. Finally we check whether pairs in $\mathcal{G}_h^i$ are more likely to establish trust relations at t_{i+1} than those in $\mathcal{G}_l^i$. We assume that $h_i(x)$ and $l_i(x)$ are the percentages of pairs in $\mathcal{G}_h^i$ and $\mathcal{G}_l^i$ establishing trust relations at time t_{i+1} at the ratio of $x\%$, respectively.

Since the percentage of pairs with trust relations is very small such as 0.0042 in Epinions, we vary x from 0.0001 to 0.01 with an incremental step of 0.0001. For each $x\%$, the average percentages of pairs establishing trust relations for higher-similarity groups, $\bar{h}_x$, and lower-similarity groups, $\bar{l}_x$, over timestamps from t_1 to t_{10}, are defined as follows,

$$\bar{h}_x = \frac{1}{10}\sum_{i=1}^{10} h_i(x), \qquad \bar{l}_x = \frac{1}{10}\sum_{i=1}^{10} l_i(x) \tag{1}$$

Let $\mathbf{h} = [\bar{h}_{0.0001}, \ldots, \bar{h}_{0.01}]$ and $\mathbf{l} = [\bar{l}_{0.0001}, \ldots, \bar{l}_{0.01}]$. With these two vectors, we also conduct a two-sample t-test on them. The null hypothesis is $H_0 : \mathbf{h} = \mathbf{l}$, and the alternative hypothesis is $H_1 : \mathbf{h} > \mathbf{l}$. The null hypothesis is rejected at significance level $\alpha = 0.01$ with p-value of $7.59e{-59}$ in Epinions. This supports that *users with higher rating similarity are more likely to establish trust relations than those with lower similarity*, which answers the second question.

Positive answers to both questions provide evidence of the existence of homophily in trust relations. With the verification of homophily in trust relations, we next study how to exploit homophily effect for trust prediction.

3. LOW-RANK MATRIX FACTORIZATION MODEL FOR TRUST PREDICTION

Low-rank matrix factorization method is widely employed in various applications such as collective filtering [8, 21] and document clustering [26, 2]. Let $\mathbf{u} = \{u_1, u_2, \ldots, u_n\}$ be the set of users where n is the number of users. $\mathbf{G} \in \mathbb{R}^{n \times n}$ is the matrix representation of trust relations where $\mathbf{G}(i,j) = 1$ if we observe that u_i trusts u_j and $\mathbf{G}(i,j) = 0$ otherwise. A few factors can influence people to establish trust relations and a user usually establishes trust relations with a small proportion of $\mathbf{u}$, resulting in $\mathbf{G}$ very sparse and low-rank, hence users can have a more compact but accurate representation in a low-rank space. The matrix factorization model seeks a low-rank representation $\mathbf{U} \in \mathbb{R}^{n \times d}$ with $d \ll n$ for $\mathbf{u}$

via solving the following optimization problem,

$$\min_{\mathbf{U},\mathbf{V}} \quad \|\mathbf{G} - \mathbf{U}\mathbf{V}\mathbf{U}^\top\|_F^2, \tag{2}$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix and $\mathbf{V} \in \mathbb{R}^{d\times d}$ captures the correlations among their low-rank representations such as $\mathbf{G}(i,j) = \mathbf{U}(i,:)\mathbf{V}\mathbf{U}^\top(j,:)$. To avoid overfitting, we add two smoothness regularizations on $\mathbf{U}$ and $\mathbf{V}$, respectively, into Eq. (2), and then we have,

$$\min_{\mathbf{U},\mathbf{V}} \quad \|\mathbf{G} - \mathbf{U}\mathbf{V}\mathbf{U}^\top\|_F^2 + \alpha\|\mathbf{U}\|_F^2 + \beta\|\mathbf{V}\|_F^2, \tag{3}$$

where α and β are non-negative and are introduced to control the capability of $\mathbf{U}$ and $\mathbf{V}$, respectively. Non-negative constraints are always applied to $\mathbf{U}$ and $\mathbf{V}$ in Eq. (3) as,

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V}} \quad & \|\mathbf{G} - \mathbf{U}\mathbf{V}\mathbf{U}^\top\|_F^2 + \alpha\|\mathbf{U}\|_F^2 + \beta\|\mathbf{V}\|_F^2 \\ s.t. \quad & \mathbf{U} \geq 0, \quad \mathbf{V} \geq 0, \end{aligned} \tag{4}$$

if the dimensionality of the latent space d is set to the number of communities, and then the low-rank representation of users $\mathbf{U}$ can be explained as an affiliation matrix, indicating the community structure of users [20].

There are several nice properties of the matrix factorization method [4, 16]: (1) many optimization methods such as gradient based methods can be applied to find a well-worked optimal solution, scaled to thousands of users with millions of trust relations; (2) it has a nice probabilistic interpretation with Gaussian noise; (3) it is very flexible and allows us to include prior knowledge such as homophily regularization, introduced in the next section.

4. MODELING HOMOPHILY FOR TRUST PREDICTION

In this section, we study how to model homophily effect in trust prediction under the low-rank matrix factorization model. After introducing homophily regularization, we present our proposed framework with its optimization method. To verify the efficiency, we present the time complexity of our framework.

4.1 Homophily Regularization

The analysis in Section 2 suggests the existence of homophily in trust relations and homophily effect indicates that users with higher similarity are more likely to establish trust relations. 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,

$$\min \quad \sum_{i=1}^{n}\sum_{j=1}^{n} \zeta(i,j)\|\mathbf{U}(i,:) - \mathbf{U}(j,:)\|_2^2, \tag{5}$$

users close to each other in the low-rank space are more likely to establish trust relations [18] 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 should be 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 her latent representation $\mathbf{U}(i,:)$ are,

$$\sum_{j=1}^{n} \zeta(i,j)\|\mathbf{U}(i,:) - \mathbf{U}(j,:)\|_2^2, \tag{6}$$

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,

$$\begin{aligned} & \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \zeta(i,j)\|\mathbf{U}(i,:) - \mathbf{U}(j,:)\|_2^2 \\ &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{d} \zeta(i,j)\big(\mathbf{U}(i,k) - \mathbf{U}(j,k)\big)^2 \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{d} \zeta(i,j)\mathbf{U}^2(i,k) \\ &\qquad - \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{d} \zeta(i,j)\mathbf{U}(i,k)\mathbf{U}(j,k) \\ &= \sum_{k=1}^{d} \mathbf{U}^\top(:,k)(\mathbf{D} - \mathcal{Z})\mathbf{U}(:,k) \\ &= Tr(\mathbf{U}^\top\mathcal{L}\mathbf{U}), \end{aligned} \tag{7}$$

where $\mathcal{L} = \mathbf{D} - \mathcal{Z}$ is the Laplacian matrix and $\mathbf{D}$ is a diagonal matrix with the i-th diagonal element $\mathbf{D}(i,i) = \sum_{j=1}^{n}\mathcal{Z}(j,i)$. $\mathcal{Z}$ is the homophily coefficient matrix among n instances, defined as,

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

Ziegler et al. pointed out that there is a strong and significant correlation between trust and user preference similarity [27]. Meanwhile, homophily effect indicates that the more similar two people are, the more likely they will establish trust relations. In the context of product view sites, user preference can be inferred from their ratings, hence homophily coefficient in this work is simply measured via rating similarity although there are other more sophisticated measures [15]. For u_i, we assume that $I(i)$ is the set of items u_i rates and R_{ij} is the rating to the j-th item from u_i. We investigate the following three widely used rating similarity measures [12] 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,

$$\zeta(i,j) = JC(u_i,u_j) = \frac{|I(i)\cap I(j)|}{|I(i)\cup I(j)|}. \tag{8}$$

- *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 5 stars while u_j gives 1 star to the j-th item. To capture different tastes from different users, we define rating similarity (RS) as,

$$\zeta(i,j) = RS(u_i, u_j) = \frac{\sum_k R_{ik} \cdot R_{jk}}{\sqrt{\sum_k R_{ik}^2}\sqrt{\sum_k R_{jk}^2}}, \quad (9)$$

  actually $RS(u_i, u_j)$ is the cosin similarity between the rating vectors of u_i and u_j, already discussed in above section 2.1.

- *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,

$$\begin{aligned}\zeta(i,j) &= PCC(u_i, u_j)\\ &= \frac{\sum_{k\in I(i)\cap I(j)}(R_{ik} - \bar{R}_i)\cdot(R_{jk} - \bar{R}_j)}{\sqrt{\sum_k (R_{ik} - \bar{R}_i)^2}\sqrt{\sum_k (R_{jk} - \bar{R}_j)^2}}, \end{aligned}\quad (10)$$

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

From these definitions of similarity functions, $JC(u_i, u_j)$ and $RS(u_i, u_j)$ range within $[0, 1]$, however, $PCC(u_i, u_j)$ ranges from -1 to 1. According to the definition of homophily coefficient, $\zeta(i,j)$ belongs to $[0, 1]$. Therefore, a mapping function $f(x) = \frac{x+1}{2}$ is applied to PCC, limiting the value of $PCC(u_i, u_j)$ within $[0, 1]$. It is easy to verify that JC, RS and PCC satisfy the three properties of homophily coefficient. Note that in this paper, we donot consider the combinations of different measures for homophily coefficient, introducing more parameters to prune and leaving it as our future work.

4.2 The Proposed Framework: hTrust

With the definition of homophily regularization, we propose a framework, hTrust, based on matrix factorization while exploiting homophily effect for trust prediction. hTrust is to solve the following optimization problem,

$$\begin{aligned}\min_{\mathbf{U},\mathbf{V}} \quad & F = \|\mathbf{G} - \mathbf{U}\mathbf{V}\mathbf{U}^\top\|_F^2\\ & + \alpha\|\mathbf{U}\|_F^2 + \beta\|\mathbf{V}\|_F^2 + \lambda Tr(\mathbf{U}^\top\mathcal{L}\mathbf{U})\\ & s.t. \quad \mathbf{U} \geq 0, \quad \mathbf{V} \geq 0. \end{aligned}\quad (11)$$

By removing constants in the objective function, Eq. (11) can be rewritten as,

$$\begin{aligned}F &= Tr(-2\mathbf{G}^\top\mathbf{U}\mathbf{V}\mathbf{U}^\top + \mathbf{U}\mathbf{V}^\top\mathbf{U}^\top\mathbf{U}\mathbf{V}\mathbf{U}^\top)\\ &\quad + \alpha Tr(\mathbf{U}\mathbf{U}^\top) + \beta Tr(\mathbf{V}\mathbf{V}^\top) + \lambda Tr(\mathbf{U}^\top\mathcal{L}\mathbf{U}). \end{aligned}\quad (12)$$

The coupling between $\mathbf{U}$ and $\mathbf{V}$ makes the problem in Eq. (11) difficult to find optimal solutions for both $\mathbf{U}$ and $\mathbf{V}$ simultaneously. In this work, we adopt an alternative optimization scheme [2] for Eq. (11), under which we update $\mathbf{U}$ and $\mathbf{V}$ alternatingly with the following updating rules,

$$\begin{aligned}\mathbf{U}(i,k) &\leftarrow \mathbf{U}(i,k)\sqrt{\frac{\mathbf{A}(i,k)}{\mathbf{B}(i,k)}},\\ \mathbf{V}(i,k) &\leftarrow \mathbf{V}(i,k)\sqrt{\frac{[\mathbf{U}^\top\mathbf{G}\mathbf{U}](i,k)}{[\mathbf{U}^\top\mathbf{U}\mathbf{V}\mathbf{U}^\top\mathbf{U} + \beta\mathbf{V}](i,k)}}, \end{aligned}\quad (13)$$

where $\mathbf{A}$ and $\mathbf{B}$ are defined as,

$$\begin{aligned}\mathbf{A} &= \mathbf{G}^\top\mathbf{U}\mathbf{V} + \mathbf{G}\mathbf{U}\mathbf{V}^\top + \lambda\mathcal{Z}\mathbf{U},\\ \mathbf{B} &= \mathbf{U}\mathbf{V}^\top\mathbf{U}^\top\mathbf{U}\mathbf{V} + \mathbf{U}\mathbf{V}\mathbf{U}^\top\mathbf{U}\mathbf{V}^\top + \alpha\mathbf{U} + \lambda\mathbf{D}\mathbf{U}. \end{aligned}\quad (14)$$

Next we will prove the correctness of the updating rules in Eq. (13) by showing that the final solution would satisfy the KKT condition. The Lagrangian function of Eq (11) is:

$$\begin{aligned}L_F &= Tr(-2\mathbf{G}^\top\mathbf{U}\mathbf{V}\mathbf{U}^\top + \mathbf{U}\mathbf{V}^\top\mathbf{U}^\top\mathbf{U}\mathbf{V}\mathbf{U}^\top)\\ &\quad + \alpha Tr(\mathbf{U}\mathbf{U}^\top) + \beta Tr(\mathbf{V}\mathbf{V}^\top) + \lambda Tr(\mathbf{U}^\top\mathcal{L}\mathbf{U})\\ &\quad - Tr(\Lambda_1\mathbf{U}) - Tr(\Lambda_2\mathbf{V}), \end{aligned}\quad (15)$$

where Λ_1 and Λ_2 are Lagrangian multipliers for non-negativity of $\mathbf{U}$ and $\mathbf{V}$, respectively.

Then we have,

$$\begin{aligned}\frac{\partial L_F}{\partial \mathbf{U}} &= 2\big(-\mathbf{G}^\top\mathbf{U}\mathbf{V} - \mathbf{G}\mathbf{U}\mathbf{V}^\top + \mathbf{U}\mathbf{V}^\top\mathbf{U}^\top\mathbf{U}\mathbf{V}\\ &\quad + \mathbf{U}\mathbf{V}\mathbf{U}^\top\mathbf{U}\mathbf{V}^\top + \alpha\mathbf{U} + \lambda\mathcal{L}\mathbf{U}\big) - \Lambda_1^\top,\\ \frac{\partial L_F}{\partial \mathbf{V}} &= 2\big(-\mathbf{U}^\top\mathbf{G}\mathbf{U} + \mathbf{U}^\top\mathbf{U}\mathbf{V}\mathbf{U}^\top\mathbf{U} + \beta\mathbf{V}\big) - \Lambda_2^\top. \end{aligned}\quad (16)$$

The KKT complementary condition is,

$$\begin{aligned}&\mathbf{U}(i,k)\Lambda_1(i,k) = 0, \quad \forall\, i \in [1,n],\ k \in [1,d]\\ &\mathbf{V}(i,k)\Lambda_2(i,k) = 0 \quad \forall\, i,\ k \in [1,d]. \end{aligned}\quad (17)$$

Let $\frac{\partial L_F}{\partial \mathbf{U}} = 0$ and $\frac{\partial L_F}{\partial \mathbf{V}} = 0$,

$$\begin{aligned}&-\mathbf{G}^\top\mathbf{U}\mathbf{V} - \mathbf{G}\mathbf{U}\mathbf{V}^\top + \mathbf{U}\mathbf{V}^\top\mathbf{U}^\top\mathbf{U}\mathbf{V}\\ &+ \mathbf{U}\mathbf{V}\mathbf{U}^\top\mathbf{U}\mathbf{V}^\top + \alpha\mathbf{U} + \lambda\mathcal{L}\mathbf{U} = \Lambda_1,\\ &-\mathbf{U}^\top\mathbf{G}\mathbf{U} + \mathbf{U}^\top\mathbf{U}\mathbf{V}\mathbf{U}^\top\mathbf{U} + \beta\mathbf{V} = \Lambda_2. \end{aligned}\quad (18)$$

Using the KKT complementary condition in Eq. (17), we have,

$$\begin{aligned}&[-\mathbf{G}^\top\mathbf{U}\mathbf{V} - \mathbf{G}\mathbf{U}\mathbf{V}^\top + \mathbf{U}\mathbf{V}^\top\mathbf{U}^\top\mathbf{U}\mathbf{V}\\ &+ \mathbf{U}\mathbf{V}\mathbf{U}^\top\mathbf{U}\mathbf{V}^\top + \alpha\mathbf{U} + \lambda\mathcal{L}\mathbf{U}](i,k)\mathbf{U}(i,k) = 0,\\ &[-\mathbf{U}^\top\mathbf{G}\mathbf{U} + \mathbf{U}^\top\mathbf{U}\mathbf{V}\mathbf{U}^\top\mathbf{U} + \beta\mathbf{V}](i,k)\mathbf{V}(i,k) = 0. \end{aligned}\quad (19)$$

It is easy to verify that the updating rules in Eq. (13) do satisfy the above KKT condition. Furthermore, since $\mathbf{G}$, $\mathcal{Z}$, $\mathbf{D}$ are nonnegative, so $\mathbf{U}$ and $\mathbf{V}$ are nonnegative during the updating process. Until now, we prove the correctness of the updating rules in Eq. (13). It can be proven that the updating rules in Eq. (13) are guaranteed to converge. Since the proof process is similar to that in [2], to save space, we omit the detailed proof of the convergence of the updating rules in Eq. (13).

4.3 The Algorithm and Time Complexity

The detailed algorithm for the proposed framework, hTrust, is shown in Algorithm 1. We construct homophily coefficient matrix in line 1. From line 4 to line 9, we alternatingly update $\mathbf{U}$ and $\mathbf{V}$ until achieving convergence. Note that in practice, Algorithm 1 will stop when reaching predefined maximal iterations or there is little change for the objective function value. After obtaining the optimal $\mathbf{U}$ and $\mathbf{V}$, $\tilde{\mathbf{G}} = \mathbf{U}\mathbf{V}\mathbf{U}^\top$ is the new low-rank representation of $\mathbf{G}$. Since $\mathbf{U}$ and $\mathbf{V}$ are non-negative, the new low-rank representation of the trust network $\tilde{\mathbf{G}}$ is non-negative. The likelihood of u_i and u_j to establish trust relation is indicated by $\tilde{\mathbf{G}}(i,j)$[4].

Algorithm 1 The Framework of Trust Prediction with Homophily Regularization

Input: $\mathbf{G}$, α, β, λ
Output: Ranking list of pairs of users

```
Construct the Homophily Coefficient Matrix Z and D
Initialize U randomly
Initialize V randomly
while Not convergent do
  Set A = G^T UV + GUV^T + λZU
  Set B = UV^T U^T UV + UVU^T UV^T + αU + λDU
  for i = 1 to n do
    for k = 1 to d do
      Update U(i,k) ← U(i,k) sqrt( A(i,k) / B(i,k) )
    end for
  end for
  for i = 1 to d do
    for k = 1 to d do
      Update V(i,k) ← V(i,k) sqrt( [U^T GU](i,k) / [U^T UVU^T U + βV](i,k) )
    end for
  end for
end while
Set G~ = UVU^T
Ranking pairs of users (e.g., <u_i, u_j>) according to G~
  (e.g., G~(i,j)) in a descending order.
```

At each iteration, the high cost of the updating rules for $\mathbf{U}$ and $\mathbf{V}$ may limit the applications of the proposed algorithm, so it is essential to analyze the time complexity and find an efficient implementation of Algorithm 1.

First we consider the time complexity of $\mathbf{A} = \mathbf{G}^\top\mathbf{U}\mathbf{V} + \mathbf{G}\mathbf{U}\mathbf{V}^\top + \lambda\mathcal{Z}\mathbf{U}$. The matrix representation of trust network $\mathbf{G}$ is very sparse thus $\mathbf{G}^\top\mathbf{U}\mathbf{V}$ and $\mathbf{G}\mathbf{U}\mathbf{V}^\top$ can be computed in $O(nd^2)$. In our studied datasets, $\mathcal{Z}$ is very sparse (by considering the top similar users, we also can obtain a sparse homophily coefficient matrix $\mathcal{Z}$ for other datasets). Therefore, the time complexity of $\lambda\mathcal{Z}\mathbf{U}$ is $O(nd^2)$.

For $\mathbf{B} = \mathbf{U}\mathbf{V}^\top\mathbf{U}^\top\mathbf{U}\mathbf{V} + \mathbf{U}\mathbf{V}\mathbf{U}^\top\mathbf{U}\mathbf{V}^\top + \alpha\mathbf{U} + \lambda\mathbf{D}\mathbf{U}$, we can calculate $\mathbf{U}\mathbf{V}^\top\mathbf{U}^\top\mathbf{U}\mathbf{V}$ by either,

$$(((\mathbf{U}\mathbf{V}^\top)\mathbf{U}^\top)\mathbf{U})\mathbf{V}, \quad or \quad \mathbf{U}(\mathbf{V}^\top((\mathbf{U}^\top\mathbf{U})\mathbf{V})) \tag{20}$$

The former takes $O(n^2d)$ operations, while the latter costs $O(nd^2)$. As $d \ll n$, the latter is much more efficient. Similarly, $\mathbf{U}\mathbf{V}\mathbf{U}^\top\mathbf{U}\mathbf{V}^\top$ should be calculated as

$$\mathbf{U}(\mathbf{V}((\mathbf{U}^\top\mathbf{U})\mathbf{V}^\top)). \tag{21}$$

Note that $\mathbf{U}^\top\mathbf{U}$ is only calculated once for $\mathbf{U}\mathbf{V}^\top\mathbf{U}^\top\mathbf{U}\mathbf{V}$ and $\mathbf{U}\mathbf{V}\mathbf{U}^\top\mathbf{U}\mathbf{V}^\top$. Since $\mathbf{D}$ is a diagonal matrix, the time complexity of $\mathbf{D}\mathbf{U}$ is $O(nd)$.

Due to the sparsity of $\mathbf{G}$, $\mathbf{U}^\top\mathbf{G}\mathbf{U}$ in the updating rule for $\mathbf{V}$ costs $O(nd^2)$. $\mathbf{U}^\top\mathbf{U}\mathbf{V}\mathbf{U}^\top\mathbf{U}$ should be computed as,

$$(\mathbf{U}^\top\mathbf{U})\mathbf{V}(\mathbf{U}^\top\mathbf{U}) \tag{22}$$

[4]The code is available at http://www.public.asu.edu/~jtang20/datasetcode/hTrust.m

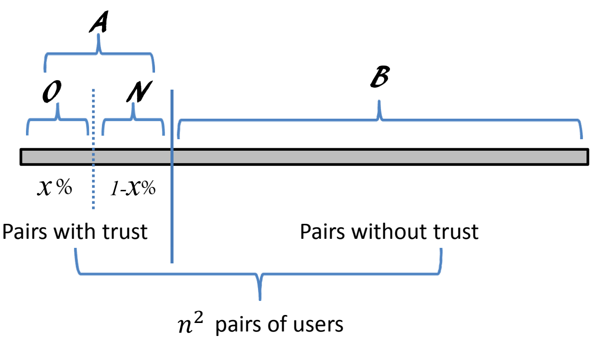


Figure 4: Separation of the dataset. $\mathcal{A}$ is the set of pairs with trust relations, sorted in chronological order, $x\%$ of which is chosen as old trust relations $\mathcal{O}$ and the remaining $1 - x\%$ as new trust relations $\mathcal{N}$ to predict, and $\mathcal{B}$ is the set of pairs without trust relations.

whose time complexity is $O(nd^2)$.

In summary, with above implementations, the overall time complexity for Algorithm 1 is $\#iterations * O(nd^2)$.

5. EXPERIMENTS

In this section, we conduct experiments to evaluate the proposed framework. After introducing the experiment settings and the evaluation metric, we compare different trust prediction methods, and then study the effect of homophily regularization and different measures of homophily coefficient on the proposed framework.

5.1 Experiment Settings

The experiment setting of the dataset is demonstrated in Figure 4. $\mathcal{A} = \{\langle u_i, u_j\rangle | \mathbf{G}(i,j) = 1\}$ is the set of pairs of users with trust relations and $\mathcal{B} = \{\langle u_i, u_j\rangle | \mathbf{G}(i,j) = 0\}$ is the set of pairs of users without trust relations. The 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 $\mathcal{O}$ and the remaining $1 - x\%$ as new trust relations $\mathcal{N}$ to predict. We remove trust relations in $\mathcal{N}$ by setting $\mathbf{G}(i,j) = 0$, $\forall\langle u_i, u_j\rangle \in \mathcal{N}$ and the new representation of $\mathbf{G}$ is the input of each predictor. x is varied as $\{50, 55, 60, 65, 70, 80, 90\}$.

We follow the common metric for unsupervised trust prediction in [9] to evaluate the performance of trust prediction. In details, each trust predictor ranks pairs in $\mathcal{B} \cup \mathcal{N}$ in decreasing order of confidence and we take the first $|\mathcal{N}|$ pairs as the set of predicted trust relations, denoting as $\mathcal{C}$. Then the prediction accuracy (PA) can be calculated as,

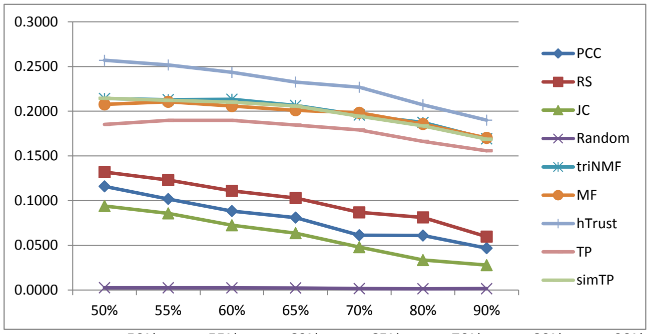
$$PA = \frac{|\mathcal{N} \cap \mathcal{C}|}{|\mathcal{N}|} \tag{23}$$

where $|\cdot|$ denotes the size of a set.

5.2 Comparison of Different Trust Predictors

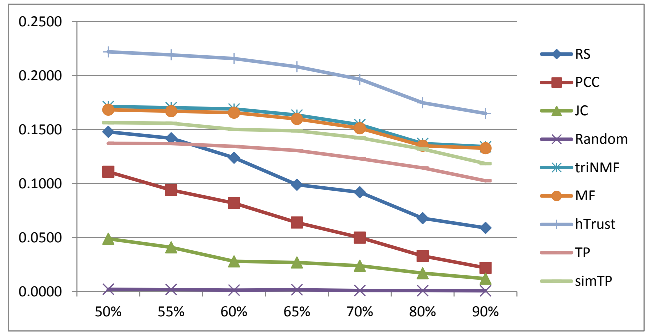
In this subsection, we compare the proposed framework with various baseline methods as follows,

- *TP:* the trust relations are inferred through trust propagation. Four atomic propagations are utilized in this



	50%	55%	60%	65%	70%	80%	90%
TP	0.1852	0.1897	0.1897	0.1845	0.1790	0.1663	0.1558
RS	0.1319	0.1230	0.1110	0.1029	0.0869	0.0813	0.0598
PCC	0.1160	0.1019	0.0884	0.0811	0.0614	0.0610	0.0469
JC	0.0940	0.0858	0.0725	0.0637	0.0480	0.0336	0.0279
simTP	0.2076	0.2105	0.2057	0.2011	0.1982	0.1857	0.1702
MF	0.2145	0.2121	0.2102	0.2057	0.1944	0.1837	0.1688
triNMF	0.2142	0.2129	0.2134	0.2064	0.1958	0.1875	0.1692
hTrust	0.2569	0.2517	0.2434	0.2326	0.2268	0.2072	0.1900
Random	0.0027	0.0026	0.0025	0.0024	0.0017	0.0015	0.0016

Figure 5: Performance Comparison for Different Trust Predictors in Epinions



	50%	55%	60%	65%	70%	80%	90%
TP	0.1374	0.1371	0.1345	0.1306	0.1231	0.1145	0.1027
RS	0.1480	0.1420	0.1240	0.0990	0.0920	0.0680	0.0590
PCC	0.1110	0.0940	0.0820	0.0640	0.0500	0.0330	0.0220
JC	0.0490	0.0410	0.0280	0.0270	0.0240	0.0170	0.0120
simTP	0.1685	0.1671	0.1657	0.1599	0.1513	0.1351	0.1328
MF	0.1564	0.1559	0.1502	0.1489	0.1424	0.1321	0.1185
triNMF	0.1714	0.1703	0.1691	0.1635	0.1546	0.1372	0.1342
hTrust	0.2220	0.2193	0.2158	0.2082	0.1966	0.1749	0.1650
Random	0.0022	0.0020	0.0014	0.0017	0.0010	0.0009	0.0008

Figure 6: Performance Comparison for Different Trust Predictors in Ciao

predictor, i.e., direct propagation, co-citation, transpose trust and trust coupling [7].

- *RS*: RS ranks the pairs of users via their rating similarity defined in Eq. (9).
- *PCC*: PCC ranks the pairs of users through Pearson Correlation Coefficient as in Eq. (10).
- *JC*: the likelihood of a pair of users to establish a trust relation is computed by Jaccard's coefficient on their rated items shown in Eq. (8).
- *simTP*: the score of a potential trust relation is estimated via a combination of trust propagation and users' rating similarity, following the basic ideas in [5, 1]. We use *RS* to measure the rating similarity.
- *MF:* it conducts a matrix factorization on the matrix representation of trust relations [26].
- *triNMF:* triNMF is a variant of our proposed method without homophily regularization as shown in Eq. (4).

Note that we do not compare our proposed framework with the methods proposed in [10, 17, 13] because: (1) these methods need additional data sources to work such as users' interaction activities; (2) homophily regularization is easy to incorporate into these methods to improve prediction performance through graph regularization; and (3) these methods are supervised methods while our proposed framework is unsupervised learning. We would like to leave the work of exploiting homophily effect for supervised methods as future work since we focus on unsupervised trust prediction in this work.

The parameters in all methods are determined through cross validation. For hTrust, we choose RS to measure the homophily coefficient and other parameters are set as $\alpha = \beta = 0.01, \lambda = 10, d = 100$. More details about the effect of homophily regularization and measures of homophily coefficient on hTrust will be discussed in later subsections. The comparison results are demonstrated in Figure 5 and Figure 6 for Epinions and Ciao, respectively.

The first observation is that with the increase of x, the performance of all methods reduces. In general, with more old trust relations , we should obtain better performance for *the same set* of new trust relations. However, in our experiments, the sets of new trust relations are different for different x%s and the difficulty of inferring new trust relations, buried in a large amount of pairs without trust relations, increases with the increase of x, supported by the trend of the performance of randomly guessing. To clarify the confusion, we conduct a validation experiment by fixing the set of new trust relations to 10% and the experimental results are shown in Table 2. Note that we only show the results of *TP*, *MF* and *hTrust*, since other methods are independent on trust networks (e.g., JC) or are variants of selected methods (e.g., *triNMF* is a variant of *hTrust*). It is clear that with the increase of new trust relations, the performance does increase in our expectation. We also note that when x is from 90 to 50, hTrust is more stable than both *TP* and *MF*. On average, the performance of *hTrust*, *TP* and *MF* relatively reduces 5.70%, 8.71% and 8.64% in Epinions, respectively, and 7.81%, 11.28% and 11.58% in Ciao, respectively. *With homophily regularization, hTrust is more robust to the sparsity problem of trust prediction.*

We have the following observations:

- The performance of RS, PCC and JC is much better than that of randomly guessing, further demonstrating the existence of homophily in trust relations. We also note that RS and PCC obtain better performance than JC. It supports that users rating the same items might have very different preferences.
- *simTP*, combining both trust propagation and rating similarity, outperforms both *TP* and similarity-based methods (RS, PCC, JC). We believe that rating infor-

Table 2: Performance of Different Predictors when the Percentage of Testing Trust Relations is Fixed to 10%

Datasets		*TP*	*MF*	*hTrust*
Epinions	50%	0.1201	0.1291	0.1578
	55%	0.1327	0.1401	0.1689
	60%	0.1335	0.1454	0.1751
	65%	0.1437	0.1555	0.1799
	70%	0.1473	0.1611	0.1821
	80%	0.1521	0.1643	0.1872
	90%	0.1558	0.1688	0.1900
Ciao	50%	0.0831	0.0988	0.1450
	55%	0.0902	0.1025	0.1496
	60%	0.0943	0.1079	0.1537
	65%	0.0957	0.1109	0.1591
	70%	0.0983	0.1132	0.1613
	80%	0.1009	0.1163	0.1649
	90%	0.1027	0.1185	0.1650

mation provides complementary information beyond trust networks for trust prediction.

- Comparing *MF* and *triNMF* with *TP*, we note that low-rank matrix factorization methods obtain better performance than trust propagation.
- Our proposed framework obtains better performance than *triNMF*. As mentioned above, *triNMF* is a variant of the proposed framework without homophily regularization, hence, these results directly demonstrate that homophily regularization can improve the performance of trust prediction. Furthermore, it also outperforms *simTP*, which also takes advantage of both existing trust relations and rating information, suggesting the effectiveness of homophily regularization in terms of capturing rating information.

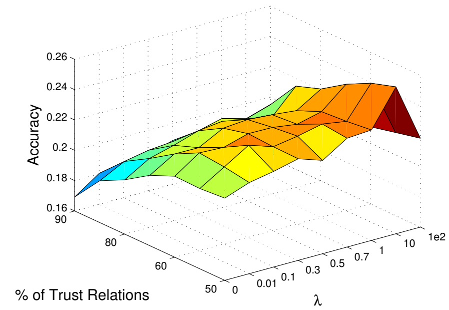
In summary, with the help of homophily regularization, the proposed framework always outperforms all the baseline methods and its variant. In the next subsection, we investigate more details about the impact of homophily regularization on the proposed framework.

5.3 Impact of Homophily Regularization

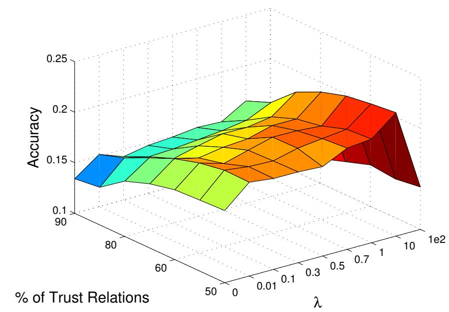
The parameter λ is introduced to control the contribution from homophily regularization for our proposed framework hTrust. Therefore we investigate the impact of homophily regularization via analyzing how the changes of λ affect the performance of hTrust in terms of the trust prediction accuracy. We vary the value of λ as $\{0, 0.01, 0.1, 0.3, 0.5, 0.7, 1, 10, 1e2\}$ and the results are shown in Figure 7(a) and Figure 7(b) for Epinions and Ciao, respectively.

In general, with the increase of λ, the performance in Epinions and Ciao shows similar patterns: first increasing, reaching its peak value and then degrading rapidly. These patterns can be used to determine the optimal value of λ for hTrust in practice. In particular, it can be observed,

- when λ is increased from 0, eliminating the impact of homophily regularization on hTrust, to 0.01, the performance improves a lot, suggesting that homophily regularization can significantly improve the performance of trust prediction.



(a) Epinions



(b) Ciao

Figure 7: Effect of Homophily Regularization

- hTrust achieves its best performance when $\lambda = 10$, further demonstrating the importance of homophily regularization in hTrust.
- From $\lambda = 10$ to $\lambda = 1e2$, the performance decreases rapidly. When λ is very large, homophily regularization dominates the learning process and the learned latent presentation is inaccurate. For example, when $\lambda \rightarrow +\infty$, we will obtain a trivial solution: all $\mathbf{U}(i,:)(1 \leq i \leq n)$ are exactly the same.

In summary, an appropriate combination of matrix factorization and homophily regularization can greatly improve the performance of trust prediction.

5.4 Impact of Homophily Coefficient

Homophily coefficient $\zeta(i,j)$ controls the distance of u_i and u_j in the latent space. In this paper, we employ three widely used measures for homophily coefficient, i.e., Jaccard's coefficient, Rating Similarity and Pearson Correlation Coefficient. To investigate the impact from homophily coefficient, we try to answer the following three questions through experiments,

- Among JC, RS and PCC, which measure of homophily coefficient is more effective?
- What is the performance of our proposed framework if we discard homophily coefficients by giving equal homophily coefficients to all pairs of users?
- If we randomly assign homophily coefficients to all pairs of users, what is the performance of our proposed framework hTrust?

Table 3: Different Measures of Homophily Coefficient. Note that $\zeta(i,j) = random$ means we randomly assign homophily coefficients ,while $\zeta(i,j) = 1$ indicates that homophily coefficients for all pairs of users are set to 1

Datasets		$\zeta(i,j) = JC(i,j)$	$\zeta(i,j) = PCC(i,j)$	$\zeta(i,j) = RS(i,j)$	$\zeta(i,j) = random$	$\zeta(i,j) = 1$
Epinions	50%	0.2382	0.2415	0.2569	0.2172	0.2192
	55%	0.2301	0.2354	0.2517	0.2153	0.2208
	60%	0.2227	0.2285	0.2434	0.2027	0.2071
	65%	0.2131	0.2196	0.2326	0.1907	0.1966
	70%	0.2019	0.2073	0.2268	0.1799	0.1856
	80%	0.1871	0.1937	0.2072	0.1558	0.1697
	90%	0.1732	0.1753	0.1900	0.1433	0.1498
Ciao	50%	0.1967	0.2098	0.2220	0.1630	0.1742
	55%	0.1941	0.2041	0.2193	0.1728	0.1721
	60%	0.1865	0.2069	0.2158	0.1585	0.1627
	65%	0.1780	0.1958	0.2082	0.1591	0.1613
	70%	0.1639	0.1820	0.1966	0.1479	0.1491
	80%	0.1441	0.1618	0.1749	0.1242	0.1304
	90%	0.1319	0.1502	0.1650	0.1214	0.1268

Table 3 demonstrates the performance of the proposed framework with different measures of homophily coefficient. In the table, $\zeta(i,j) = random$ means we randomly assign homophily coefficients within $[0,1]$,while $\zeta(i,j) = 1$ indicates that homophily coefficients for all pairs of users are set to 1. From the table, we observe answers to the questions proposed at the beginning of this subsection:

- 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. It suggests that different users might have different tastes to the same item.
- Compared to JC, RS and PCC, the performance degrades when we assign equal homophily coefficient, i.e.,$\zeta(i,j) = 1$. JC, RS and PCC can improve the performance of trust prediction.
- Among the measures of homophily coefficient, most of the time, random assignment of homophily coefficients obtains the worst performance. Homophily coefficient should not be a random value.

After answering the three questions, we can conclude that (1) JC, RS and PCC can help our framework obtain better performance; (2) we cannot either discard homophily coefficient from our framework or simply use some random values to denote homophily coefficients.

6. RELATED WORK

Trust plays an important role in helping online users collect reliable information and trust prediction, inferring unknown trust relations among pairs of users, attracts more and more attention in recent years [7, 5, 14, 10, 17, 13, 1]. Existing trust prediction methods can be roughly divided into two categories: unsupervised trust prediction [7, 5, 14] and supervised trust prediction [10, 17, 13].

Most existing unsupervised trust prediction methods are based on trust propagation. Several atomic propagations are proposed in [7] such as direct propagation, cocitation propagation, transpose propagation and trust coupling propagation. It also discusses the propagation of distrust and develop a formal framework of trust propagation schemes. Various properties of trust such as transitivity, composability and asymmetry are discussed by Golbeck in [5] and based on these properties, algorithms for inferring binary and continuous trust values from trust networks are proposed. The continuous trust inference algorithm, *TidalTrust*, leverages the path length from the source to sink and various properties of continuous ratings [5]. Trust propagation based methods strongly depend on existing trust relations among users and they might fail when existing trust relations are sparse. In [1], rating similarity is exploited to enrich traditional trust propagation methods. This work demonstrates that predicting trust is more successful for pairs of users that are similar to each other if we combine the topology of the trust network with rating similarity.

Supervised trust prediction methods first construct features from available sources and then train a binary classifier based on these features by considering the existence of trust relations as labels. In [10], a taxonomy is developed to systematically organize an extensive set of features for predicting trust relations. The features include user and interaction factors. User factors contain rater-related, writer-related, or commenter-related. The latter captures various interactions between the users. Viet-An Nguyen et al. [17] proposes various trust prediction models based on a well-studied Trust Antecedent Framework used in management science, capturing the three following factors: ability, benevolence and integrity. Each factor is approximated through a set of quantitative features. For example, the features for integrity are called trustworthiness, equal to the number of trust statements the user receives while ability are the features that compute the average rating given by a rater to the reviews written by a particular reviewer and the number of reviews rated by the rater. In [13] , various features based on writer-reviewer interactions are extracted and used in personalized and cluster-based classification methods. As mentioned above, these methods have inherent limitations. The huge disproportion of pairs of users with (positive samples) and without relations (negative samples) makes the classification problem extremely unbalanced and the performance of these methods are sensitive to the sampled negative samples [23]. Trust has multiple facets and people place trust differently on different people [21]. Considering heterogeneous trust relations and their evolution can improve various trust-related applications such as trust prediction and trust-aware recommendation [22].

7. CONCLUSION

In this paper, we study the problem of exploiting homophily effect for trust prediction. First we conduct experiments on datasets from real-world product review sites to demonstrate the existence of homophily in trust relations. Homophily regularization is then introduced to capture homophily effect in trust relations. An unsupervised framework is proposed, incorporating low-rank matrix factorization and homophily regularization. Extensive experiments are conducted to evaluate the proposed framework on real-world trust relation datasets and the experimental results demonstrate the effectiveness of our proposed framework as well as the role of homophily regularization for trust prediction.

There are several interesting directions for future work. In the current work, homophily regularization is utilized under an unsupervised scenario. As mentioned above, homophily regularization can be easily extended for supervised learning methods via graph regularization. Hence one direction of our future work is to examine the effect of homophily regularization on supervised trust predictors. Previous work demonstrated that homophily is widely observed in other relations such as following relations in Twitter; another direction of our future work is to generalize homophily regularization to other relations. Users might change their preferences over time, indicating their homophily coefficients might evolve. In the future, we will further study homophily regularization with temporal dynamics for trust prediction.

Acknowledgments

The work is, in part, supported by ARO (#025071) and NSF (#IIS-1217466).

8. REFERENCES

[1] P. Borzymek, M. Sydow, and A. Wierzbicki. Enriching trust prediction model in social network with user rating similarity. In *International Conference on Computational Aspects of Social Networks*, 2009.

[2] C. Ding, T. Li, and M. Jordan. Nonnegative matrix factorization for combinatorial optimization: Spectral clustering, graph matching, and clique finding. In *ICDM*, 2008.

[3] Y. Hu, A. John, F. Wang,and S. Kambhampati. ET-LDA: Joint Topic Modeling for Aligning Events and their Twitter Feedback. In *AAAI*, 2012.

[4] D. Dunlavy, T. Kolda, and E. Acar. Temporal link prediction using matrix and tensor factorizations. *TKDD*, 2011.

[5] J. Golbeck. Generating predictive movie recommendations from trust in social networks. *Trust Management*, 2006.

[6] J. Golbeck. Trust and nuanced profile similarity in online social networks. *ACM Transactions on the Web*, 3(4):1–33, 2009.

[7] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *WWW*, 2004.

[8] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD*, 2008.

[9] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 2007.

[10] H. Liu, E. Lim, H. Lauw, M. Le, A. Sun, J. Srivastava, and Y. Kim. Predicting trusts among users of online communities: an epinions case study. In *EC*, 2008.

[11] Y. Lu, P. Tsaparas, A. Ntoulas, and L. Polanyi. Exploiting social context for review quality prediction. In *WWW*, 2010.

[12] H. Ma, D. Zhou, C. Liu, M. Lyu, and I. King. Recommender systems with social regularization. In *WSDM*, 2011.

[13] N. Ma, E. Lim, V. Nguyen, A. Sun, and H. Liu. Trust relationship prediction using online product review data. In *Proceeding of the 1st ACM international workshop on Complex networks meet information & knowledge management*, pages 47–54. ACM, 2009.

[14] P. Massa and P. Avesani. Controversial users demand local trust metrics: An experimental study on epinions.com community. In *Proceedings of the National Conference on Artificial Intelligence*, 2005.

[15] M. McPherson, L. Smith-Lovin, and J. Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 2001.

[16] A. Menon and C. Elkan. Link prediction via matrix factorization. *Machine Learning and Knowledge Discovery in Databases*, 2011.

[17] V. Nguyen, E. Lim, J. Jiang, and A. Sun. To trust or not to trust? predicting online trusts using trust antecedent framework. In *ICDM*, 2009.

[18] S. Oyama, K. Hayashi, and H. Kashima. Cross-temporal link prediction. In *ICDM*, 2011.

[19] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD*, 2002.

[20] J. Tang, X. Wang, and H. Liu. Integrating Social Media Data for Community Detection. In *Modeling and Mining Ubiquitous Social Media*, 2012.

[21] J. Tang, H. Gao, and H. Liu. mTrust: Discerning multi-faceted trust in a connected world. In *WSDM*, 2012.

[22] J. Tang, H. Gao, H. Liu, and A. Das Sarma. eTrust: Understanding trust evolution in an online world. In *KDD*, 2012.

[23] D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A. Barabási. Human mobility, social ties, and link prediction. In *KDD*,2011.

[24] J. Weng, E. Lim, J. Jiang, and Q. He. Twitterrank: finding topic-sensitive influential twitterers. In *WSDM*, 2010.

[25] L. Peng, R. Yu, K. Selcuk, and X. Wang. Object and Combination Shedding Schemes for Adaptive Media Workflow Execution. In *TKDE*, 2010.

[26] S. Zhu, K. Yu, Y. Chi, and Y. Gong. Combining content and link for classification using matrix factorization. In *SIGIR*, 2007.

[27] C. Ziegler and J. Golbeck. Investigating interactions of trust and interest similarity. *Decision Support Systems*, 2007.