MATRI: A Multi-Aspect and Transitive Trust Inference Model

Yuan Yao
State Key Laboratory for Novel Software Technology, China
yyao@smail.nju.edu.cn

Hanghang Tong
City College, CUNY, USA
tong@cs.ccny.cuny.edu

Xifeng Yan
University of California at Santa Barbara, USA
xyan@cs.ucsb.edu

Feng Xu
State Key Laboratory for Novel Software Technology, China
xf@nju.edu.cn

Jian Lu
State Key Laboratory for Novel Software Technology, China
lj@nju.edu.cn

ABSTRACT

Trust inference, which is the mechanism to build new pair-wise trustworthiness relationship based on the existing ones, is a fundamental integral part in many real applications, e.g., e-commerce, social networks, peer-to-peer networks, etc. State-of-the-art trust inference approaches mainly employ the *transitivity* property of trust by propagating trust along connected users (a.k.a. trust propagation), but largely ignore other important properties, e.g., prior knowledge, multi-aspect, etc.

In this paper, we propose a multi-aspect trust inference model by exploring an equally important property of trust, i.e., the *multi-aspect* property. The heart of our method is to view the problem as a recommendation problem, and hence opens the door to the rich methodologies in the field of collaborative filtering. The proposed multi-aspect model directly characterizes multiple *latent* factors for each trustor and trustee from the locally-generated trust relationships. Moreover, we extend this model to incorporate the prior knowledge as well as trust propagation to further improve inference accuracy. We conduct extensive experimental evaluations on real data sets, which demonstrate that our method achieves *significant* improvement over several existing benchmark approaches. Overall, the proposed method (MATRI) leads to *26.7% - 40.7%* improvement over its best known competitors in prediction accuracy; and up to *7 orders of magnitude* speedup with *linear* scalability.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database applications—*Data mining*

Keywords

Trust inference; transitivity property; multi-aspect property; latent factors; prior knowledge

1. INTRODUCTION

Trust is essential to reduce uncertainty and boost collaborations in many real-world applications including social networks [39], e-commerce [11], peer-to-peer networks [12], semantic Web [25], etc. In these applications, trust inference is widely used as the mechanism to build trust among unknown users. Typically, trust inference takes as its input the existing trust ratings that are locally-generated through direct interactions, and outputs an estimated trustworthiness score from a trustor to an unknown trustee. This estimated trustworthiness score indicates to what extent the trustor could expect the trustee to perform a given action.

The basic assumption behind most of the existing trust inference methods is the *transitivity* property of trust [19], which is rooted in the social structural balance theory [4]. This property essentially means that if *Alice* trusts *Bob* and *Bob* trusts *Carol*, *Alice* might also trust *Carol* to some extent. These methods (see Section 6 for a review), referred to as *trust propagation* models as a whole, have been widely studied and successfully applied in many real-world settings [8, 39, 19, 15, 9, 22].

In addition to transitivity, a few trust inference models explore another equally important property, that is, the *multi-aspect* of trust [6, 27]. The basic assumption behind the multi-aspect methods is that trust is the composition of multiple factors, and different users may have different preferences for these factors. For example, in e-commerce, some users might care more about the factor of delivering time, whereas others give a higher weight to the factor of product price. However, the existing multi-aspect trust inference methods [26, 36, 31, 28] require as its input some side information (e.g., the delivering time as well as user's preference for it, etc) in addition to the locally-generated trust ratings, and therefore become infeasible in many trust networks where such side information may not be available.

Another limitation in existing trust inference models is that they tend to ignore some important prior knowledge (e.g., trust bias) during the inference procedure. It was discovered in sociology a long time ago that *trust bias* is an integral part in the final trust decision [30]. Nonetheless, it was not until the very recent years did the computer science community begin to incorporate the trust bias into the inference process. For example, a recent work [23] models trustor bias as the propensity of a trustor to trust others.

In this paper, we aim to integrate all these important properties, including transitivity, multi-aspect and prior knowledge, to maximally boost the inference accuracy. We start by proposing a multi-aspect trust inference model. The heart of our method is to view the problem as a recommendation problem, and hence opens the door to the rich methodologies in the field of collaborative filtering. The proposed multi-aspect model directly characterizes multiple *latent* factors for each trustor and trustee from the locally-generated trust relationships. Based on that, we propose to incorporate the prior knowledge as *specified* aspects and automatically learn the relative weights between latent and specified factors. Finally, we extend

WWW 2013, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2035-1/13/05.

this model to incorporate trust propagation to further improve inference accuracy.

To summarize, the main contributions of this paper are as follows:

(1) *Trust Models.* To the best of our knowledge, this is the first work to (a) integrate transitivity, multi-aspect and prior knowledge into one single trust inference model; and (b) directly characterize the multi-aspect trustworthiness relationship solely based on locally-generated trust ratings. It can admit the rich methodologies from collaborative filtering. It is flexible to model the prior knowledge as specified factors and further learn their relative weights.

(2) *Performance Improvements.* We conducted extensive experimental evaluations on two widely used benchmark data sets, and empirically observed significant performance improvements in both effectiveness and efficiency. In terms of prediction accuracy, our MATRI outperforms the best known existing methods by 26.7% - 40.7%. By pre-computation, our MATRI is much faster in terms of on-line response, achieving up to 7 orders of magnitude speedup. Finally, the pre-computation stage itself of the proposed MATRI scales *linearly* wrt the size of the input data set, indicating that it is suitable for large data sets.

The rest of the paper is organized as follows. Section 2 presents the definition of the trust inference problem. Section 3 describes our optimization formulation for the problem defined in the previous section and shows how to incorporate prior knowledge and trust propagation. Section 4 presents the inference algorithm to solve the formulation. Section 5 presents experimental results. Section 6 reviews related work. Section 7 concludes the paper.

2. PROBLEM DEFINITION

In this section, we formally define our trust inference problem. Table 1 lists the main symbols we use throughout the paper.

Table 1: Symbols.

Symbol	Definition and Description
$\mathbf{T}$	the partially observed trust matrix
$\mathbf{F}, \mathbf{G}$	the characterized trustor and trustee matrices
$\mathbf{F}_0, \mathbf{G}_0$	the sub-matrix of $\mathbf{F}$ and $\mathbf{G}$
$\mathbf{T}'$	the transpose of matrix $\mathbf{T}$
$\mathbf{T}(i, j)$	the element at the i^{th} row and j^{th} column of $\mathbf{T}$
$\mathbf{T}(i, :)$	the i^{th} row of matrix $\mathbf{T}$
$\mathcal{K}$	the set of observed trustor-trustee pairs in $\mathbf{T}$
μ	the global bias
$\mathbf{x}$	the vector of trustor bias
$\mathbf{y}$	the vector of trustee bias
$\mathbf{x}(i)$	the i^{th} element of vector $\mathbf{x}$
$\mathbf{z}_{ij}$	the vector of propagation elements for trustor-trustee pair (i, j)
n	the number of users
p, r	the number of bias and latent factors
s	total number of factors, $s = p + r$
t	the maximum propagation step
α_i	the weights/coefficients for bias factors
β_j	the weights/coefficients for propagation elements
u, v	the trustor and the trustee
m	the maximum iteration number
ξ	the threshold to terminate the iteration

Following conventions, we use bold capital letters for matrices, and bold lower case letters for vectors. For example, we use a partially observed matrix $\mathbf{T}$ to model the locally-generated trust relationships, where the existing/observed trust relationships are represented as non-zero trust ratings and non-existing/unobserved relationships are represented as '?'. As for the observed trust rating, we represent it as a real number between 0 and 1 (a higher rating means more trustworthiness). We use calligraphic font $\mathcal{K}$ to denote the set of observed trustor-trustee indices in $\mathbf{T}$. Similar to Matlab, we also denote the i^{th} row of matrix $\mathbf{T}$ as $\mathbf{T}(i, :)$, and the transpose of a matrix with a prime. In addition, we denote the number of users as n and the number of characterized factors as s. Without loss of generality, we assume that the goal of our trust model is to infer the unseen trust relationship from the user u to another user v, where u is the trustor and v is the unknown trustee to u.

Based on these notations, we first define the basic trust inference problem as follows:

PROBLEM 1. *The Basic Trust Inference Problem*

Given: *an $n \times n$ partially observed trust matrix $\mathbf{T}$, a trustor u, and a trustee v, where $1 \leqslant u, v \leqslant n$ $(u \neq v)$ and $\mathbf{T}(u, v)$ = '?';*

Find: *the estimated trustworthiness score $\hat{\mathbf{T}}(u, v)$.*

In the above problem definition, given a trustor-trustee pair, the only information we need as input is the locally-generated trust ratings (i.e., the partially observed matrix $\mathbf{T}$). The goal of trust inference is to infer the new trust ratings (i.e., unseen/unobserved trustworthiness scores in the partially observed matrix $\mathbf{T}$) by collecting the knowledge from existing trust relationships. In this paper, we assume that we can access such existing trust relationships. For instance, these relationships could be collected by central servers in a centralized environment like eBay, or by individuals in a distributed environment like EigenTrust [12]. How to collect these trust relationships is out of the scope of this work.

As mentioned before, one of our goals is to capture the multi-aspect property of trust. In this paper, we propose a multi-aspect model for such trust inference in Problem 1. That is, we want to infer an $n \times s$ *trustor matrix* $\mathbf{F}$ whose element indicates to what extent the corresponding person trusts others wrt a specific aspect/factor. Similarly, we want to infer another $n \times s$ *trustee matrix* $\mathbf{G}$ whose element indicates to what extent the corresponding person is trusted by others wrt a specific aspect/factor. Such trustor and trustee matrices are in turn used to infer the unseen trustworthiness scores. Based on the basic trust inference problem, we define the multi-aspect trust inference problem as follows:

PROBLEM 2. *The Multi-Aspect Trust Inference Problem*

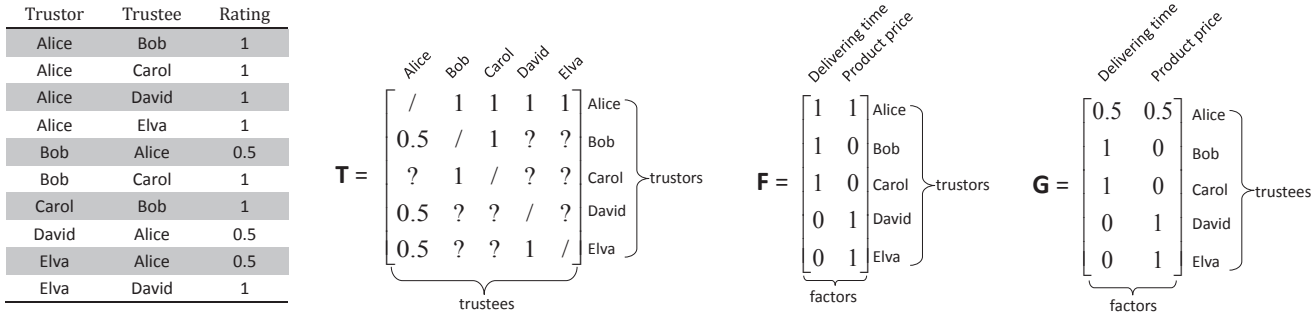
Given: *an $n \times n$ partially observed trust matrix $\mathbf{T}$, the number of factors s, a trustor u, and a trustee v, where $1 \leqslant u, v \leqslant n$ $(u \neq v)$ and $\mathbf{T}(u, v)$ = '?';*

Find: *(1) an $n \times s$ trustor matrix $\mathbf{F}$ and an $n \times s$ trustee matrix $\mathbf{G}$; (2) the estimated trustworthiness score $\hat{\mathbf{T}}(u, v)$.*

2.1 An Illustrative Example

To further illustrate our multi-aspect trust inference problem (Problem 2), we give an intuitive example as shown in Fig. 1.

In this example, we observe several locally-generated pair-wise trust relationships between five users (e.g., '*Alice*', '*Bob*', '*Carol*', '*David*', and '*Elva*') as shown in Fig. 1(a). Each observation contains a trustor, a trustee, and a numerical trust rating from the trustor to the trustee. We then model these observations as a 5×5 partially observed matrix $\mathbf{T}$ (see Fig. 1(b)) where $\mathbf{T}(i, j)$ is the trust rating



(a) The observed locally-generated pair-wise trust relationships

(b) The partially observed trust matrix **T**

(c) The inferred trustor matrix **F** and trustee matrix **G**

Figure 1: An illustrative example for multi-aspect trust inference problem.

from the i^{th} user to the j^{th} user if the rating is observed and $\mathbf{T}(i, j) =$ '?' otherwise. Notice that we do not consider self-ratings and thus represent the diagonal elements of **T** as '/'. By setting the number of factors $s = 2$, our goal is to infer two 5×2 matrices **F** and **G** (see Fig. 1(c)) from the input matrix **T**. Each row of the two matrices is for the corresponding user, and each column of the matrices represents a certain aspect/factor in trust inference (e.g., 'delivering time', 'product price', etc). For example, we can see that *Alice* trusts others strongly wrt both 'delivering time' and 'product price' (based on **F**), and she is in turn moderately trusted by others wrt these two factors (based on **G**). On the other hand, both *Bob* and *Carol* put more emphasis on the delivering time, while *David* and *Elva* care more about the product price.

Once **F** and **G** are inferred, we can use these two matrices to estimate the unseen trustworthiness scores (i.e., the '?' elements in **T**). For instance, the trustworthiness from *Carol* to *Alice* can be estimated as $\hat{\mathbf{T}}(3, 1) = \mathbf{F}(3, :)\mathbf{G}(1, :)' = 0.5$. This estimation is reasonable because *Carol* has the same preference as *Bob* and the trustworthiness score from *Bob* to *Alice* is also 0.5.

In the next two sections, we will mainly focus on (1) how to infer **F** and **G**; and (2) how to incorporate prior knowledge (i.e., trust bias) and trust transitivity (i.e., trust propagation) based on the partially observed input matrix **T**.

3. THE OPTIMIZATION FORMULATION

In this section, we present our optimization formulation to integrate all the three important properties in trust inference, including multi-aspect, prior knowledge (i.e., trust bias) and trust transitivity (i.e., trust propagation). We start with the basic form to capture the multi-aspect of trust; and then show how to incorporate trust bias and four groups of trust propagation. Finally, we discuss some generalizations of our formulation.

3.1 The Basic Formulation

In this work, we adopt optimization method to solve the trust inference problem defined in the previous section. Formally, Problem 2 can be formulated as the following optimization problem:

$$\min_{\mathbf{F},\mathbf{G}} \sum_{(i,j)\in\mathcal{K}} (\mathbf{T}(i, j) - \mathbf{F}(i, :)\mathbf{G}(j, :)')^2 + \lambda\|\mathbf{F}\|_{fro}^2 + \lambda\|\mathbf{G}\|_{fro}^2 \qquad (1)$$

where λ is a regularization parameter; $\|\mathbf{F}\|_{fro}$ and $\|\mathbf{G}\|_{fro}$ are the Frobenius norm of the trustor and trustee matrices, respectively.

By this formulation, it aims to minimize the squared error on the set of observed trust ratings. Notice that in Eq. (1), we have two additional regularization terms ($\|\mathbf{F}\|_{fro}^2$ and $\|\mathbf{G}\|_{fro}^2$) to improve the solution stability. The parameter $\lambda \geq 0$ controls the amount of such regularization. Based on the resulting **F** and **G** of the above equation, the unseen trustworthiness score $\hat{\mathbf{T}}(u, v)$ can then be estimated by $\mathbf{F}(u, :)$ and $\mathbf{G}(v, :)$ as:

$$\hat{\mathbf{T}}(u, v) = \mathbf{F}(u, :)\mathbf{G}(v, :)' \qquad (2)$$

A Collaborative Filtering Metaphor. As mentioned in introduction, we view the trust inference problem as a recommendation problem. To be specific, in the trust matrix **T**, if we treat its rows (i.e., trustors) as 'users'; its columns (i.e., trustees) as 'items'; and its entries (i.e., trustworthiness scores) as 'ratings', the optimization problem in Eq. (1) resembles the same form as that of so-called factorization-based collaborative filtering [13]. This viewpoint opens the door to the rich methodologies in collaborative filtering to capture the multi-aspect of trust.

3.2 Incorporating Trust Bias

The formulation in Eq. (1) can naturally incorporate some prior knowledge such as trust bias into the inference procedure. In this paper, we explicitly consider the following three types of trust bias (i.e., $p = 3$ where p is the number of bias factors): *global bias*, *trustor bias*, and *trustee bias*, although other types of bias can be incorporated in a similar way.

Global bias: The global bias represents the average level of trust in the community. The intuition behind this is that users tend to rate optimistically in some reciprocal environments (e.g., e-commerce) while they are more conservative in others (e.g., security-related applications). As a result, it might be useful to take such global bias into account and we model it as a scalar μ.

Trustor bias: The trustor bias is based on the observation that some trustors tend to generously give higher trust ratings than others. This bias reflects the propensity of a given trustor to trust others, and it may vary a lot among different trustors. Accordingly, we can model the trustor bias as vector $\mathbf{x}$ with $\mathbf{x}(i)$ indicating the trust propensity of the i^{th} trustor.

Trustee bias: The third type of bias aims to characterize the fact that some trustees might have relatively higher capability in terms of being trusted than others. Similar to the second type of bias, we model this type of bias as vector $\mathbf{y}$, where $\mathbf{y}(j)$ indicates the overall capability of the j^{th} trustee compared to the average.

Each of these three types of bias can be represented as a *specified* factor for our model, respectively. By incorporating such bias into

Eq. (1), we have the following formulation:

$$\min_{\mathbf{F},\mathbf{G}} \sum_{(i,j)\in\mathcal{K}} (\mathbf{T}(i,j) - \mathbf{F}(i,:)\mathbf{G}(j,:)')^2 + \lambda\|\mathbf{F}\|_{fro}^2 + \lambda\|\mathbf{G}\|_{fro}^2$$

$$\text{Subject to:} \quad \mathbf{F}(:,1) = \mu\mathbf{1},\ \mathbf{G}(:,1) = \alpha_1\mathbf{1}/\sqrt{n} \quad \text{(global bias)}$$
$$\mathbf{F}(:,2) = \mathbf{x},\ \mathbf{G}(:,2) = \alpha_2\mathbf{1}/\sqrt{n} \quad \text{(trustor bias)}$$
$$\mathbf{F}(:,3) = \alpha_3\mathbf{1}/\sqrt{n},\ \mathbf{G}(:,3) = \mathbf{y} \quad \text{(trustee bias)} \quad (3)$$

where α_1, α_2, and α_3 are the weights of bias that we need to estimate based on the existing trust ratings.

In addition to these three specified factors, we refer to the remaining factors in the trustor and trustee matrices as *latent* factors. Let us define two $n\times r$ sub-matrices of $\mathbf{F}$ and $\mathbf{G}$ for the latent factors. That is, we define $\mathbf{F}_0 = \mathbf{F}(:, 4:s)$ and $\mathbf{G}_0 = \mathbf{G}(:, 4:s)$, where each column of $\mathbf{F}_0$ and $\mathbf{G}_0$ corresponds to one latent factor and r is the number of latent factors. With this notation, we have the following equivalent form of Eq. (3):

$$\min_{\mathbf{F}_0,\mathbf{G}_0,\boldsymbol{\alpha}} \sum_{(i,j)\in\mathcal{K}} (\mathbf{T}(i,j) - (\boldsymbol{\alpha}'[\mu, \mathbf{x}(i), \mathbf{y}(j)]' + \mathbf{F}_0(i,:)\mathbf{G}_0(j,:)'))^2$$
$$+\lambda\|\mathbf{F}_0\|_{fro}^2 + \lambda\|\mathbf{G}_0\|_{fro}^2 + \lambda\|\boldsymbol{\alpha}\|^2 \quad (4)$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \alpha_3]'$.

Recall that in this paper, we aim to perform trust inference only using the partially observed trust matrix $\mathbf{T}$. Therefore, we estimate the parameters (μ, $\mathbf{x}$ and $\mathbf{y}$) of the trust bias as follows:

$$\begin{cases} \mu = \sum_{(i,j)\in\mathcal{K}} \mathbf{T}(i,j)/|\mathcal{K}| \\ \mathbf{x}(i) = \sum_{j,(i,j)\in\mathcal{K}} \mathbf{T}(i,j)/|row_i| - \mu \\ \mathbf{y}(j) = \sum_{i,(i,j)\in\mathcal{K}} \mathbf{T}(i,j)/|col_j| - \mu \end{cases} \quad (5)$$

where $|row_i|$ is the number of the observed elements in the i^{th} row of $\mathbf{T}$, and $|col_j|$ is the number of the observed elements in the j^{th} column of $\mathbf{T}$.

3.3 Incorporating Trust Propagation

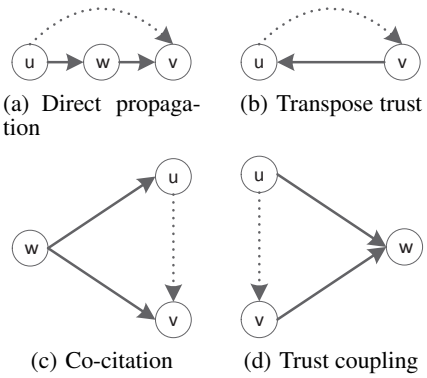
We next describe how to incorporate trust propagation into the model. We consider the following four groups of trust propagation operators defined in [8]: *direct propagation*, *transpose trust*, *co-citation*, and *trust coupling*.

Direct propagation: Direct propagation is probably the most intuitive way to propagate trust as shown in Fig. 2(a). The basic operator in the figure presents the two-step propagation and it can be generalized to multiple steps. We define the first group of $(t-1)$ propagation elements in the matrix form as $\mathbf{T}^2, \mathbf{T}^3, ..., \mathbf{T}^t$, where t is the largest propagation step.

Transpose trust: The second operator is the transpose trust as shown in Fig. 2(b). This operator indicates that user v's trust on user u can cause some level of trust in the opposite direction. This group of t propagation elements can be represented in the matrix form as $\mathbf{T}'$, $(\mathbf{T}')^2$, $(\mathbf{T}')^3$, ..., $(\mathbf{T}')^t$.

Co-citation: Co-citation is found to be very powerful to predict trust and distrust in the Epinions website. As shown in Fig. 2(c), co-citation means that if two users u and v are both trusted by another user w, then u might also trust v to some extent. Based on the transitive closure computation, we can represent this group of propagation elements as: $(\mathbf{T}'\mathbf{T}), (\mathbf{T}'\mathbf{T})^2, (\mathbf{T}'\mathbf{T})^3, ..., (\mathbf{T}'\mathbf{T})^t$.

Trust coupling: Fig. 2(d) shows the trust coupling operator, which means that if two users both trust another user, they might also trust each other. Similar to co-citation, we represent the fourth group of propagation elements as $(\mathbf{T}\mathbf{T}'), (\mathbf{T}\mathbf{T}')^2, (\mathbf{T}\mathbf{T}')^3, ..., (\mathbf{T}\mathbf{T}')^t$.



(a) Direct propagation (b) Transpose trust

(c) Co-citation (d) Trust coupling

Figure 2: The four propagation operators. The solid lines indicate existing trust relationships, and the dotted lines indicate propagated trust.

Altogether, we have generated $(4t-1)$ trust propagation matrices, with each corresponding entry measuring one specific trust propagation between the two corresponding users, respectively. For example, $\mathbf{T}^t(i,j)$ measures direct propagation from user i to user j after t steps, and $(\mathbf{T}\mathbf{T}')(i,j)$ quantifies the one-step trust coupling effect between user i and user j, etc. If we further stack all these $(4t-1)$ entries into a propagation vector $\mathbf{z}_{ij}$ for the given user pair (i,j), we have the following form when we incorporate both trust bias and trust propagation into Eq. (1):

$$\min_{\mathbf{F}_0,\mathbf{G}_0,\boldsymbol{\alpha},\boldsymbol{\beta}} \sum_{(i,j)\in\mathcal{K}} (\mathbf{T}(i,j) - (\boldsymbol{\alpha}'[\mu, \mathbf{x}(i), \mathbf{y}(j)]' + \boldsymbol{\beta}'\mathbf{z}_{ij}$$
$$+\mathbf{F}_0(i,:)\mathbf{G}_0(j,:)'))^2 + \lambda\|\mathbf{F}_0\|_{fro}^2 + \lambda\|\mathbf{G}_0\|_{fro}^2$$
$$+\lambda\|\boldsymbol{\alpha}\|^2 + \lambda\|\boldsymbol{\beta}\|^2 \quad (6)$$

where $\mathbf{z}_{ij}$ is the vector of propagation elements for the trustor-trustee pair (i,j), $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \alpha_3]'$ is the weight vector for bias, and $\boldsymbol{\beta} = [\beta_1, \beta_2, ..., \beta_{4t-1}]'$ is the weight vector for trust propagation.

Notice that there is no coefficient before $\mathbf{F}_0(i,:)\mathbf{G}_0(j,:)'$ as it will be automatically absorbed into $\mathbf{F}_0$ and $\mathbf{G}_0$ in our iterative algorithm. Once we have inferred all the parameters (i.e., $\mathbf{F}_0$, $\mathbf{G}_0$, $\boldsymbol{\alpha}$, and $\boldsymbol{\beta}$) of Eq. (6), the unseen trustworthiness score $\hat{\mathbf{T}}(u,v)$ can be immediately estimated as:

$$\hat{\mathbf{T}}(u,v) = \mathbf{F}_0(u,:)\mathbf{G}_0(v,:)' + \boldsymbol{\alpha}'[\mu, \mathbf{x}(u), \mathbf{y}(v)]' + \boldsymbol{\beta}'\mathbf{z}_{uv} \quad (7)$$

3.4 Discussions and Generalizations

We further present some discussions and generalizations of our optimization formulation.

First, it is worth pointing out that our formulation in Eq. (1) differs from the standard matrix factorization (e.g., SVD) as in the objective function, we try to minimize the square loss *only* on those observed trust pairs. This is because the majority of trust pairs are missing from the input trust matrix $\mathbf{T}$. As mentioned before, our basic problem setting in Eq. (1) is conceptually similar to the standard collaborative filtering, as in both cases, we aim to fill in missing values in a partially observed matrix (trustor-trustee matrix vs. user-item matrix). Indeed, if we fix the coefficients $\alpha_1 = \alpha_2 = \alpha_3 = 1$ and $\beta_1 = \beta_2 = ... = \beta_{4t-1} = 0$ in Eq. (6), it is reduced to the collaborative filtering algorithm in [14]. Our formulation in Eq. (6) goes beyond the standard collaborative filtering by (1) incorporating two other important properties in trust inference (i.e., bias and transitivity); and (2) learning their relative weights ($\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$). Our experimental evaluations show that such subtle treatments are cru-

cial and they lead to further performance improvement over these existing techniques.

Second, although our model is a subjective trust inference metric where different trustors may form different opinions on the same trustee [22], as a side product, the proposed model can also be used to infer an objective, unique trustworthiness score for each trustee. For example, this objective trustworthiness score can be computed based on the trustee matrix $\mathbf{G}$. We will compare this feature of the proposed model with a well studied objective trust inference metric EigenTrust [12] in the experimental evaluation section.

Finally, we would like to point out that our formulation is flexible and can be generalized to other settings. For instance, our current formulation adopts the square loss function in the objective function. In other words, we implicitly assume that the residuals of the pair-wise trustworthiness scores follow a Gaussian distribution, and in our experimental evaluations, we found it works well. Nonetheless, our upcoming proposed algorithm can be generalized to *any* Bregman divergence in the objective function. Also, we can naturally incorporate some additional constraints (e.g., non-negativity, sparseness, etc) in the trustor and trustee matrices. After we infer all the parameters (e.g., the coefficients for the bias and propagation, and the trustor and trustee matrices, etc), we use a linear combination to compute the trustworthiness score $\hat{\mathbf{T}}(u, v)$. We can also generalize this linear form to other non-linear combinations, such as the logistic function. For the sake of clarity, we skip the details of such generalizations in the paper.

4. THE PROPOSED MATRI ALGORITHM

In this section, we present the proposed algorithm (MATRI) to solve the trust inference problem in Eq. (6), followed by some effectiveness and efficiency analysis.

4.1 The MATRI Algorithm

Unfortunately, the optimization problem in Eq. (6) is not jointly convex wrt the coefficients ($\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$) and the trustor/trustee matrices ($\mathbf{F}_0$ and $\mathbf{G}_0$) due to the coupling between them. Therefore, instead of seeking for a global optimal solution, we try to find a local minima by alternatively updating the coefficients and the trustor/trustee matrices while fixing the other.

4.1.1 Sub-routine 1: updating the trustor/trustee matrices

First, let us consider how to update the trustor/trustee matrices ($\mathbf{F}_0$ and $\mathbf{G}_0$) when we fix the coefficients ($\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$). For clarity, we define an $n \times n$ matrix $\mathbf{P}$ as follows:

$$\mathbf{P}(i,j) = \begin{cases} \mathbf{T}(i,j) - (\boldsymbol{\alpha}'[\mu, \mathbf{x}(i), \mathbf{y}(j)]' + \boldsymbol{\beta}'\mathbf{z}_{ij}) & \text{if } (i,j) \in \mathcal{K} \\ \text{`?'} & \text{otherwise} \end{cases} \quad (8)$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are some fixed constants, and '?' means the rating is unknown.

Based on the above definition, Eq. (6) can be simplified (by ignoring some constant terms) as:

$$\min_{\mathbf{F}_0, \mathbf{G}_0} \sum_{(i,j)\in\mathcal{K}} (\mathbf{P}(i,j) - \mathbf{F}_0(i,:)\mathbf{G}_0(j,:)')^2 + \lambda\|\mathbf{F}_0\|_{fro}^2 + \lambda\|\mathbf{G}_0\|_{fro}^2 \quad (9)$$

Therefore, updating the trustor/trustee matrices when we fix the coefficients unchanged becomes a standard matrix factorization problem for missing values. Many existing algorithms (e.g., [14, 21, 2]) can be plugged in to solve Eq. (9). In our experiment, we found the so-called alternating strategy, where we recursively update one of the two trustee/trustor matrices while keeping the other matrix fixed, works best and thus recommend it in practice. A brief skeleton of the algorithm is shown in Alg. 1, and the detailed algorithm are presented in our technical report [38].

Algorithm 1 updateMatrix($\mathbf{P}$, r).

Input: The $n \times n$ matrix $\mathbf{P}$, and the latent factor size r
Output: The $n \times r$ trustor matrix $\mathbf{F}_0$, and the $n \times r$ trustee matrix $\mathbf{G}_0$

1: $[\mathbf{F}_0, \mathbf{G}_0]$ = `alternatingFactorization`($\mathbf{P}$, r);
2: **return** $[\mathbf{F}_0, \mathbf{G}_0]$;

Algorithm 2 computePropagation($\mathbf{T}$, l, t).

Input: The $n \times n$ matrix trust $\mathbf{T}$, the latent factor size l, and the maximum propagation step t
Output: The propagation vector $\mathbf{z}_{ij}$ for all $(i,j) \in \mathcal{K}$

1: $[\mathbf{L}, \mathbf{R}]$ = `updateMatrix`($\mathbf{T}$, l);
2: **for** each $(i,j) \in \mathcal{K}$ **do**
3: compute $\mathbf{z}_{ij}$ by Eq. (10);
4: **end for**
5: **return** $[\mathbf{z}_{ij}]$ $(i,j) \in \mathcal{K}$;

4.1.2 Sub-routine 2: computing trust propagation

Directly computing the propagation vector $\mathbf{z}_{ij}(i,j) \in \mathcal{K}$ is computationally inefficient as it involves the multiplications of matrices of $n \times n$. To address this issue, we propose the following procedure (Alg. 2) to compute the trust propagation vectors. In Alg. 2, we first factorize the input trust matrix into two low rank matrices $\mathbf{L}, \mathbf{R}$ (step 1); and use them as the base to compute the trust propagation vectors. By doing so, we only need to compute the matrix power or multiplications of $l \times l$, where $l \ll n$.

Notice that in step 1, instead of the standard SVD, we call Alg. 1 to get the two low rank matrices. In this way, we implicitly fill in the missing values in the partially observed matrix $\mathbf{T}$ before performing the propagation. This has the additional advantage to mitigate the sparsity or coverage problem in trust inference [20] where some trustor and trustee might not be connected with each other.

$$\begin{cases} \mathbf{T}^t(i,j) &= \mathbf{L}(i,:)(\mathbf{R}'\mathbf{L})^{t-1}\mathbf{R}(j,:)' \\ (\mathbf{T}')^t(i,j) &= \mathbf{R}(i,:)(\mathbf{L}'\mathbf{R})^{t-1}\mathbf{L}(j,:)' \\ (\mathbf{T}'\mathbf{T})^t(i,j) &= \mathbf{R}(i,:)((\mathbf{L}'\mathbf{L})(\mathbf{R}'\mathbf{R}))^{t-1}(\mathbf{L}'\mathbf{L})\mathbf{R}(j,:)' \\ (\mathbf{T}\mathbf{T}')^t(i,j) &= \mathbf{L}(i,:)((\mathbf{R}'\mathbf{R})(\mathbf{L}'\mathbf{L}))^{t-1}(\mathbf{R}'\mathbf{R})\mathbf{L}(j,:)' \end{cases} \quad (10)$$

4.1.3 Sub-routine 3: updating the coefficients

Here, we consider how to update the coefficients ($\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$) when we fix the trustor/trustee matrices.

If we fix the trustor and trustee matrices ($\mathbf{F}_0$ and $\mathbf{G}_0$) and let:

$$\mathbf{P}(i,j) = \begin{cases} \mathbf{T}(i,j) - \mathbf{F}_0(i,:)\mathbf{G}_0(j,:)' & \text{if } (i,j) \in \mathcal{K} \\ \text{`?'} & \text{otherwise} \end{cases} \quad (11)$$

Eq. (6) can then be simplified (by dropping constant terms) as:

$$\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} \sum_{(i,j)\in\mathcal{K}} (\mathbf{P}(i,j) - (\boldsymbol{\alpha}'[\mu, \mathbf{x}(i), \mathbf{y}(j)]' + \boldsymbol{\beta}'\mathbf{z}_{ij}))^2 + \lambda\|\boldsymbol{\alpha}\|^2 + \lambda\|\boldsymbol{\beta}\|^2 \quad (12)$$

To simplify the description, let us introduce another scalar k to index each pair (i,j) in the observed trustor-trustee pairs $\mathcal{K}$, that is, $(i,j) \in \mathcal{K} \rightarrow k = \{1, 2, ..., |\mathcal{K}|\}$. Let $\mathbf{b}$ denote a vector of length $|\mathcal{K}|$ with $\mathbf{b}(k) = \mathbf{P}(i,j)$. We also define a $|\mathcal{K}| \times (4t+2)$ matrix $\mathbf{A}$ as: $\mathbf{A}(k,1) = \mu$, $\mathbf{A}(k,2) = \mathbf{x}(i)$, $\mathbf{A}(k,3) = \mathbf{y}(j)$, $\mathbf{A}(k, 4:4t+2) = \mathbf{z}'_{ij}$, $(k = 1, 2, ..., |\mathcal{K}|)$.

Algorithm 3 MATRI($\mathbf{T}$, $\mathcal{K}$, r, l, t, u, v).

Input: The $n \times n$ partially observed trust matrix $\mathbf{T}$, the set of observed trustor-trustee pairs $\mathcal{K}$, the latent factor size r, the low rank l for trust propagation, the maximum propagation step t, trustor u, and trustee v

Output: The estimated trustworthiness score $\hat{\mathbf{T}}(u, v)$

Pre-computation stage:

1: compute bias: $[\mu, \mathbf{x}, \mathbf{y}]$ = `computeBias`($\mathbf{T}$) by Eq. (5);
2: compute propagation: $\mathbf{z}_{ij}$ = `computePropagation`($\mathbf{T}, l, t$), $(i, j) \in \mathcal{K}$;
3: initialize $\alpha_1 = \alpha_2 = \alpha_3 = 1, \beta_1 = \beta_2 = ... = \beta_{4t-1} = 0$;
4: **while** not convergent **do**
5: **for** each $(i, j) \in \mathcal{K}$ **do**
6: $\mathbf{P}(i, j) = \mathbf{T}(i, j) - (\boldsymbol{\alpha}'[\mu, \mathbf{x}(i), \mathbf{y}(j)]' + \boldsymbol{\beta}'\mathbf{z}_{ij})$;
7: **end for**
8: $[\mathbf{F}_0, \mathbf{G}_0]$ = `updateMatrix`($\mathbf{P}$, r);
9: **for** each $(i, j) \in \mathcal{K}$ **do**
10: $\mathbf{P}(i, j) = \mathbf{T}(i, j) - \mathbf{F}_0(i, :)\mathbf{G}_0(j :,)'$;
11: **end for**
12: $[\boldsymbol{\alpha}, \boldsymbol{\beta}]$ = `updateCoefficient`($\mathbf{P}$, μ, $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}_{ij}$) by Eq. (13);
13: **end while**

On-line query response stage:

14: **return** $\hat{\mathbf{T}}(u, v) = \mathbf{F}_0(u, :)\mathbf{G}_0(v, :)' + \boldsymbol{\alpha}'[\mu, \mathbf{x}(u), \mathbf{y}(v)]' + \boldsymbol{\beta}'\mathbf{z}_{uv}$;

Then, the coefficients ($\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$) can be updated by solving the following ridge regression problem, which is equivalent to Eq. (12):

$$\boldsymbol{\gamma} = [\boldsymbol{\alpha}; \boldsymbol{\beta}] = \texttt{argmin}_{\boldsymbol{\gamma}} \|\mathbf{b} - \mathbf{A}\boldsymbol{\gamma}\|^2 + \lambda\|\boldsymbol{\gamma}\|^2 \quad (13)$$

4.1.4 *Putting everything together:* MATRI

Putting everything together, we propose Alg. 3 for the trust inference problem in Eq. (6). The algorithm first computes trust bias (step 1) and trust propagation (step 2). Next, after an initialization step (step 3), the algorithm begins the alternating procedure (Step 4-13). At each iteration, it first fixes the coefficients ($\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$), and updates the trustor matrix $\mathbf{F}_0$ and trustee matrix $\mathbf{G}_0$ (step 5-8). Next, the algorithm fixes $\mathbf{F}_0$ and $\mathbf{G}_0$, and uses ridge regression in Eq. (13) to update the coefficients $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ (step 9-12). We use the following criteria to terminate the alternating procedure: either the L_2 norm between successive estimates of both $\mathbf{F}_0$ and $\mathbf{G}_0$ is below our threshold ξ or the maximum iteration step m is reached. Finally, the algorithm outputs the estimated trustworthiness score from the given trustor u to the trustee v using Eq. (7).

It is worth pointing out that Step 1-13 in the algorithm can be pre-computed and their results (including $\mathbf{F}_0$, $\mathbf{G}_0$, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, μ, $\mathbf{x}$, $\mathbf{y}$, $\mathbf{L}$,$\mathbf{R}$,$\mathbf{L}'\mathbf{R}$,$\mathbf{R}'\mathbf{L}$,$\mathbf{L}'\mathbf{L}$ and $\mathbf{R}'\mathbf{R}$) can be stored in the pre-computational or off-line stage. When an on-line trust inference request arrives, the proposed MATRI only needs to apply Step 14 to return the inference result, which only requires a constant time.

4.2 Algorithm Analysis

Here, we briefly analyze the effectiveness and efficiency of our algorithm.

The effectiveness of the proposed MATRI algorithm can be summarized in Lemma 1, which says that overall, it finds a local minima solution. Given that the original optimization problem in Eq. (6) is not jointly convex wrt the coefficients ($\boldsymbol{\alpha}, \boldsymbol{\beta}$) and the trustor/trustee matrices ($\mathbf{F}_0$ and $\mathbf{G}_0$), such a local minima is acceptable in practice.

LEMMA 1. **Effectiveness of MATRI**. *Fixing the propagation vector $\mathbf{z}_{ij}$, Alg. 3 finds a local minima for the optimization problem in Eq.* (6)*.*

PROOF. Omitted for brevity. □

The time complexity of the proposed MATRI is summarized in Lemma 2, which says that MATRI (1) requires *constant* time for on-line query response (step 14) and (2) scales *linearly* wrt the number of users and the number of the observed trustor-trustee pairs in the pre-computational stage (step 1-13).

LEMMA 2. **Time Complexity of MATRI**. *Fixing r, l and t as constants, (P1) Alg. 3 requires $O(nm+|\mathcal{K}|m)$ time for pre-computation, where m is the maximum iteration number in Alg. 3; and (P2) Alg. 3 requires $O(1)$ for on-line query response.*

PROOF. Omitted for brevity. □

The space complexity of MATRI is summarized in Lemma 3, which says that MATRI requires *linear* space wrt the number of users and the number of the observed trustor-trustee pairs.

LEMMA 3. **Space Complexity of MATRI**. *Fixing r, l and t as constants, Alg. 3 requires $O(|\mathcal{K}| + n)$ space.*

PROOF. Omitted for brevity. □

5. EXPERIMENTAL EVALUATION

In this section, we present experimental evaluations, after we introduce the data sets. All the experiments are designed to answer the following questions:

- *Effectiveness*: How accurate is the proposed MATRI for trust inference?
- *Efficiency*: How fast is the proposed MATRI? How does it scale?

5.1 Data Sets Description

Many existing trust inference models design specific simulation studies to verify the underlying assumptions of the corresponding inference models. Here, we focus on two widely used real, benchmark data sets in order to compare the performance of different trust inference models.

The first data set is *advogato*[1]. It is a trust-based social network for open source developers. To allow users to certify each other, the network provides 4 levels of trust assertions, i.e., '*Observer*', '*Apprentice*', '*Journeyer*', and '*Master*'. These assertions can be mapped into real numbers which represent the degree of trust. To be specific, we map '*Observer*', '*Apprentice*', '*Journeyer*', and '*Master*' to 0.1, 0.4, 0.7, and 0.9, respectively (a higher value means more trustworthiness).

The second data set is *PGP* (short for Pretty Good Privacy) [9]. *PGP* adopts the concept of 'web of trust' to establish a decentralized model for data encryption and decryption. Similar to *advogato*, the web of trust in *PGP* data set contains 4 levels of trust as well. In our experiments, we also map them to 0.1, 0.4, 0.7, and 0.9, respectively.

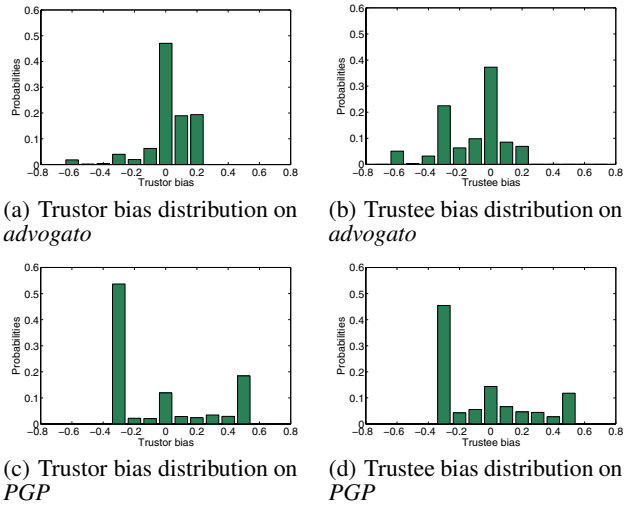
Table 2 summarizes the basic statistics of the two resulting partially observed trust matrices $\mathbf{T}$. Notice that for the *advogato* data set, it contains six different snapshots, i.e., *advogato-1*, *advogato-2*,..., *advogato-6*, etc. We use the largest snapshot (i.e., *advogato-6*) in the following unless otherwise stated.

Fig. 3 summarizes the distributions of trustor bias and trustee bias. As we can see, many users in *adovogato* perform averagely

[1]http://www.trustlet.org/wiki/Advogato_dataset.

Table 2: High level statistics of *advogato* and *PGP* data sets.

Data set	Nodes	Edges	Avg. degree	Avg. clustering [34]	Avg. diameter [17]	Date
advogato-1	279	2,109	15.1	0.45	4.62	2000-02-05
advogato-2	1,261	12,176	19.3	0.36	4.71	2000-07-18
advogato-3	2,443	22,486	18.4	0.31	4.67	2001-03-06
advogato-4	3,279	32,743	20.0	0.33	4.74	2002-01-14
advogato-5	4,158	41,308	19.9	0.33	4.83	2003-03-04
advogato-6	5,428	51,493	19.0	0.31	4.82	2011-06-23
PGP	38,546	317,979	16.5	0.45	7.70	2008-06-05



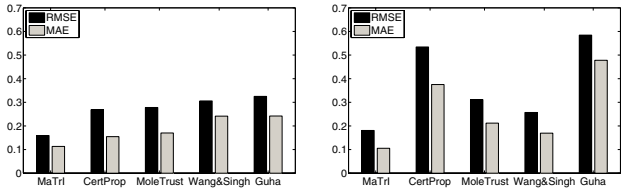
(a) Trustor bias distribution on *advogato*

(b) Trustee bias distribution on *advogato*

(c) Trustor bias distribution on *PGP*

(d) Trustee bias distribution on *PGP*

Figure 3: The distributions of trustor bias and trustee bias.



(a) *advogato* data set

(b) *PGP* data set

Figure 4: Comparisons with subjective trust inference models. Lower is better. The proposed MATRI significantly outperforms all the other existing models wrt both RMSE and MAE on both data sets.

Table 3: Performance gain analysis of MATRI. Smaller is better. Both trust propagation and trust bias further improve trust inference accuracy.

RMSE/MAE	advogato	PGP
Best known competitor	0.269 / 0.155	0.257 / 0.169
Basic form	0.256 / 0.194	0.265 / 0.155
Basic form + propagation	0.174 / 0.124	0.214 / 0.124
Basic form + bias	0.168 / 0.119	0.189 / 0.116
MATRI	0.159 / 0.113	0.181 / 0.105

on trusting others and being trusted by others. On the other hand, a considerable part of *PGP* users are cautiously trusted by others, and even more users tend to rate others conservatively. The global bias for *advogato* (0.668) is much higher than that for *PGP* (0.384). This also suggests that the security-related *PGP* network is a more conservative environment than the developer-based *advogato* network.

5.2 Effectiveness Results

We use both *advogato* (i.e., *advogato-6*) and *PGP* for effectiveness evaluations. For both data sets, we hide a randomly selected sample of 500 observed trustor-trustee pairs as the test set, and apply the proposed MATRI as well as other existing methods on the remaining data set to infer the trustworthiness scores for those hidden pairs. To evaluate and compare the accuracy, we report both the root mean squared error (RMSE) and the mean absolute error (MAE) between the estimated and the true trustworthiness scores. Both RMSE and MAE are measured on the 500 hidden pairs in the test set. We set $r = l = 10$, $m = 10$, and $\xi = 10^{-6}$ in our experiments unless otherwise stated. For the maximum propagation step t, we fix it to 6 due to the "six-degree separation".

(A) Comparisons with Existing Subjective Trust Inference Methods. We first compare the effectiveness of MATRI with several benchmark trust propagation models, including *CertProp* [9], *MoleTrust* [22], *Wang&Singh* [32, 33], and *Guha* [8]. For all these subjective methods, the goal is to infer a pair-wise trustworthiness score (i.e., to what extent the user u trusts another user v).

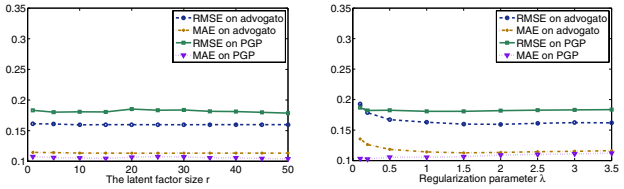
The result is shown in Fig. 4. We can see that the proposed MATRI significantly outperforms all the other trust inference models wrt both RMSE and MAE on both data sets. For example, on *advogato* data set, our MATRI improves the best existing method (CertProp) by 40.7% in RMSE and by 26.7% in MAE. As for *PGP* data set, the proposed MATRI improves the best existing method (Wang&Singh) by 29.6% in RMSE and by 37.8% in MAE. Overall, the proposed MATRI leads to 26.7% - 40.7% improvement over these best known competitors in prediction accuracy. The results suggest that multi-aspect of trust indeed plays a very important role in the inference process.

(B) Performance Gain Analysis of MATRI. Let us take a close look at where the performance gain of the proposed MATRI comes from. Recall that in the proposed MATRI, we aim to integrate the three important properties of trust, that is, *multi-aspect*, *trust bias* and *trust propagation*. We next analyze how each of these properties improves the trust inference accuracy. The result is shown in Table 3. In Table 3, 'Basic form' only considers multi-aspect of trust by setting the coefficients for trust bias as well as those for trust propagation as 0; 'Basic form + propagation' ignores the trust bias; 'Basic form + bias' ignores the trust propagation; and MATRI is the proposed method that integrates all three properties. We also show the result of the best known competitors, i.e., CertProp for *advogato* and Wang&Singh for *PGP*, in the table for comparison.

As we can see from Table 3, the performance of 'Basic form' which only considers the multi-aspect property is already close to the best known competitors. When trust propagation and trust bias

Table 4: Comparisons with *SVD*, *HCD* [10], and *KBV* [14]. Smaller is better. MATRI performs best.

RMSE/MAE	advogato	PGP
SVD	0.629 / 0.579	0.447 / 0.306
HCD	0.269 / 0.219	0.314 / 0.216
KBV	0.179 / 0.125	0.217 / 0.133
MATRI	0.159 / 0.113	0.181 / 0.105



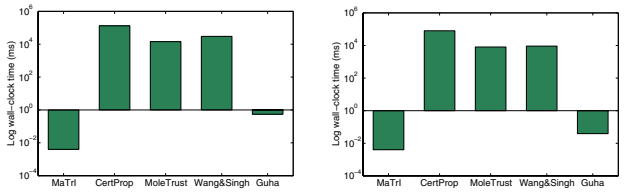
(a) RMSE and MAE of MATRI wrt r. We fix $r = 10$ for both *advogato* and *PGP*.

(b) RMSE and MAE of MATRI wrt λ. We fix $\lambda = 0.1$ for *advogato* and $\lambda = 1.0$ for *PGP*.

Figure 5: The sensitivity evaluations. MATRI is robust wrt both parameters.

Table 5: Comparisons with *EigenTrust*. Smaller is better. MATRI is better than EigenTrust wrt both RMSE and MAE on both data sets.

RMSE/MAE	advogato	PGP
EigenTrust	0.700 / 0.653	0.519 / 0.371
MATRI	0.290 / 0.203	0.349 / 0.280



(a) Wall-clock time on *advogato* data set

(b) Wall-clock time on *PGP* data set

Figure 6: Speed comparison. MATRI is much faster than all the other methods.

are incorporated, both of them significantly improve trust inference accuracy. For example, on *advogato* data set, trust propagation helps to obtain 32.0% and 36.1% improvements in RMSE and MAE, respectively. Further, trust bias improves RMSE and MAE by 8.6% and 8.9%, respectively. This result confirms our hypothesis that in addition to multi-aspect, both trust propagation and trust bias also play important roles in trust inference.

(C) Comparisons with Existing Matrix Factorization Methods. We also compare MATRI with some existing matrix factorization methods: *SVD*, the low rank approximation algorithm [10] for link sign prediction (referred to as *HCD*), and the collaborative filtering algorithm [14] for recommender systems (referred to as *KBV*).

The result is shown in Table 4. As we can see from the table, MATRI again performs best on both data sets. *SVD* performs poorly as it treats all the unobserved trustor-trustee pairs as zero elements in the trust matrix $\mathbf{T}$. MATRI outperforms *HCD* as *HCD* was essentially tailored to predict the *binary* trust/distrust relationship and it ignored the other two important properties (i.e., trust bias and trust propagation). MATRI also outperforms *KBV*. For example, MATRI improves *KBV* by 11.5% in RMSE and by 16.5% in MAE on *PGP* data set. As mentioned before, *KBV* can be viewed as a special case of the proposed MATRI if we (1) fix all the bias coefficients as 1*s* and (2) ignore the trust propagation. This result indicates that by (1) incorporating the trust propagation and (2) simultaneously learning the relative weights of propagation and trust bias, MATRI leads to further performance improvement.

(D) Sensitivity Evaluations. We also conduct a parametric study for MATRI. The first parameter is the latent factor size r. We can observe from Fig. 5(a) that, in general, both RMSE and MAE stay stable wrt r. The second parameter in MATRI is the regularization coefficient λ. As we can see from Fig. 5(b), both RMSE and MAE stay stable on *advogato*, while they decrease when λ increases up to 1.0 and stay stable after $\lambda > 1.0$ on *PGP*. Based on these results, we conclude that MATRI is robust wrt its parameters. For all the other results we report in the paper, we simply fix $r = 10$, $\lambda = 0.1$ for *advogato*, and $\lambda = 1.0$ for *PGP*.

(E) Comparisons with Existing Objective Trust Inference Methods. Although our MATRI is a subjective trust inference metric, as a side product, it can also be used to infer an objective trustworthiness score for each trustee. To this end, we set $r = 1$ in MATRI algorithm, ignore the trust propagation vectors $\mathbf{z}_{ij}$, and aggregate the resulting trustee matrix/vector $\mathbf{G}_0$ with the bias (the global bias μ and the trustee bias $\mathbf{y}$). We compare the result with a widely-cited objective trust inference model *EigenTrust* [12] in Table 5. As we can see, MATRI outperforms EigenTrust in terms of both RMSE and MAE on both data sets. For example, on *advogato* data set, MATRI is 58.6% and 68.9% better than EigenTrust wrt RMSE and MAE, respectively.

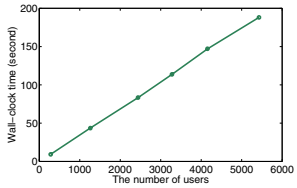
5.3 Efficiency Results

For efficiency experiments, we report the average wall-clock time. All the experiments were run on a machine with two 2.4GHz Intel Cores and 4GB memory.

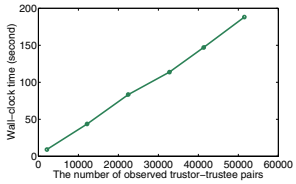
(A) Speed Comparison. We first compare the on-line response of MATRI with *CertProp*, *MoleTrust*, *Wang&Singh*, and *Guha*. Again, we use the *advogato-6* snapshot and *PGP* in this experiment, and the result is shown in Fig. 6. Notice that the y-axis is in the logarithmic scale.

We can see from the figure that the proposed MATRI is much faster than all the alternative methods on both data sets. For example, MATRI is up to 32,000,000x faster than CertProp. This is because once we have inferred the trustor/truestee matrices as well as the coefficients for the bias and propagation, it only takes *constant* time for MATRI to output the trustworthiness score. Among all the alternative methods, Guha is the most efficient. This is because its main workload can also be completed in advance. However, the pre-computation of Guha needs additional $O(n^2)$ space as the model fills nearly all the missing elements in the trust matrix, making it unsuitable for large data sets. In contrast, our MATRI only requires $O(|\mathcal{K}| + n)$ space.

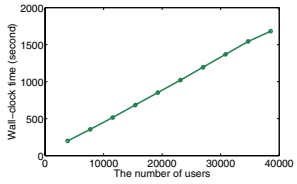
(B) Scalability. Finally, we present the scalability result of MATRI by reporting the wall-clock time of the pre-computational stage (i.e., Step 1-13 in Alg. 3). For *advogato* data set, we directly report the results on all the six snapshots (i.e., *advogato-1*, ..., *advogato-6*). For *PGP*, we use its subsets to study the scalability. The result is shown in Fig. 7, which is consistent with the complexity analysis in Section 4.2. As we can see from the figure, MATRI scales linearly wrt to both n and $|\mathcal{K}|$, indicating that it is suitable for large-scale applications.



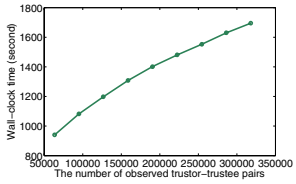
(a) Wall-clock time vs. n on *advogato*



(b) Wall-clock time vs. $|\mathcal{K}|$ on *advogato*



(c) Wall-clock time vs. n on *PGP*



(d) Wall-clock time vs. $|\mathcal{K}|$ on *PGP*

Figure 7: Scalability of the proposed MATRI. MATRI scales linearly wrt the data size (n and $|\mathcal{K}|$).

6. RELATED WORK

In this section, we briefly review related work, including trust propagation models, multi-aspect trust inference models, etc.

Trust Propagation Models. To date, a large body of trust inference models are based on trust propagation where trust is propagated along connected users in the trust network, i.e., the web of locally-generated trust ratings. Based on the interpretation of trust propagation, we further categorize these models into two classes: *path interpretation* and *component interpretation*.

In the first category of path interpretation, trust is propagated along a path from the trustor to the trustee, and the propagated trust from multiple paths can be combined to form a final trustworthiness score. For example, Wang et al. [32, 33] as well as Hang et al. [9] propose operators to concatenate trust along a path and aggregate trust from multiple paths. Liu et al. [18] argue that not only trust values but social relationships and recommendation role are important for trust inference. In contrast, there is no explicit concept of paths in the second category of component interpretation. Instead, trust is treated as random walks on a graph or on a Markov chain [25]. Examples of this category include [8, 22, 39, 15].

The proposed MATRI integrates the trust propagation with two other important properties, i.e., the multi-aspect of trust and trust bias. In addition, our multi-aspect model offers a natural way to speed up on-line query response; as well as to mitigate the sparsity or coverage problem in trust inference where some trustor and trustee might not be connected with each other - both are known limitations with the current trust propagation models [37, 20].

Multi-Aspect Trust Inference Models. Social scientists have explored the multi-aspect property of trust for several years [27]. In computer science, there also exist a few trust inference models that *explicitly* explore the multi-aspect property of trust. For example, Xiong and Liu [36] model the value of the transaction in trust inference; Wang and Wu [31] take competence and honesty into consideration; Tang et al. [28] model aspect as a set of products that are similar to each other under product review sites; Sabater and Sierra [26] divide trust in e-commerce environment into three aspects: price, delivering time, and quality.

However, all these existing multi-aspect trust inference methods require some additional side information other than the locally-generated trust ratings, such as the value of transaction, user's preference, product categories, etc. These methods become infeasible when such side information is not available. In contrast, MATRI directly characterizes the multi-aspect of trust solely based on the locally-generated trust ratings; and therefore it has a broader applicability.

Prior Knowledge in Trust Inference. In sociology, it was discovered a long time ago that certain prior knowledge, e.g., *trust bias*, is an integral part in the final trust decision [30]. Nonetheless, this important aspect has been largely ignored in most of the existing trust inference models. One exception is the work by Nguyen et al. [24], which learns the importance of several trust bias related features derived from a social trust framework. Recently, Mishra et al. [23] propose an iterative algorithm to compute trust bias. Different from these existing works, our focus is to incorporate various types of trust bias as specified factors/aspects to increase the accuracy of trust inference.

Collaborative Filtering vs. Trust Inference. Multi-aspect or low rank approximation methods have been extensively studied in collaborative filtering [1, 14, 21]. These work provides rich methodologies to capture the multi-aspect of trust by viewing the trust inference as a collaborative filtering problem. The proposed MATRI takes one step further by (1) incorporating trust bias and trust propagation; and (2) learning their relative weights.

On the application side, the goal of collaborative filtering is to predict users' flavors of items. It is interesting to point out that (1) on one hand, trust between users could help to predict the flavors as we may give a higher weight to the recommendations provided by trusted users; (2) on the other hand, trust itself might be affected by the similarity of flavors since users usually trust others with a similar taste [7]. Although out of the scope of this paper, using recommendation to further improve trust inference accuracy might be an interesting topic for future work.

Other Related Work. The concept of stereotype for trust inference is studied by Liu et al. [20] and Burnett et al. [3]. These methods learn the stereotypes from the user profiles of the trustees that the trustor has interacted with, and then use these stereotypes to reflect the trustor's first impression about unknown trustees. Several other work focuses on trust dynamics [29] and the relationship between trust and similarity [7, 35]. There are also some recent work on using link prediction approaches to predict the *binary* trust/distrust relationship [16, 5, 10].

7. CONCLUSION

In this paper, we have proposed an effective trust inference model (MATRI). The basic idea is to leverage the multi-aspect property of trust by characterizing several aspects/factors for each trustor and trustee based on the existing trust relationships. The proposed MATRI incorporates the trust propagation and prior knowledge (i.e., trust bias); and further learns their relative weights. By integrating all these important properties, our experimental evaluations on real benchmark data sets show that it leads to significant improvement in prediction accuracy. The proposed MATRI is also nimble - it is up to 7 orders of magnitude faster than the existing methods in the on-line query response, and in the meanwhile it enjoys the linear scalability for the pre-computational stage in both time and space. Future work includes investigating the capability of MATRI to address the distrust as well as the trust dynamics.

8. ACKNOWLEDGMENTS

We would like to thank the valuable suggestions from Jiliang Tang as well as the anonymous reviewers. This work is supported by the National Natural Science Foundation of China (No. 61021062,

61073030, 61100037), the National 863 Program of China (No. 2012AA011205), and the National 973 Program of China (No. 2009CB320702). It is partly supported by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053, the U.S. Defense Advanced Research Projects Agency (DARPA) under Agreement Number W911NF-12-C-0028, and the National Science Foundation under Grant No. IIS-1017415.

9. REFERENCES

[1] R. Bell, Y. Koren, and C. Volinsky. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *KDD*, pages 95–104. ACM, 2007.

[2] A. Buchanan and A. Fitzgibbon. Damped newton algorithms for matrix factorization with missing data. In *CVPR*, volume 2, pages 316–322, 2005.

[3] C. Burnett, T. Norman, and K. Sycara. Bootstrapping trust evaluations through stereotypes. In *AAMAS*, pages 241–248, 2010.

[4] D. Cartwright and F. Harary. Structural balance: a generalization of heider's theory. *Psychological Review*, 63(5):277–293, 1956.

[5] K. Chiang, N. Natarajan, A. Tewari, and I. Dhillon. Exploiting longer cycles for link prediction in signed networks. In *CIKM*, pages 1157–1162, 2011.

[6] D. Gefen. Reflections on the dimensions of trust and trustworthiness among online consumers. *ACM SIGMIS Database*, 33(3):38–53, 2002.

[7] J. Golbeck. Trust and nuanced profile similarity in online social networks. *ACM Transactions on the Web*, 3(4):12, 2009.

[8] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *WWW*, pages 403–412. ACM, 2004.

[9] C.-W. Hang, Y. Wang, and M. P. Singh. Operators for propagating trust and their evaluation in social networks. In *AAMAS*, pages 1025–1032, 2009.

[10] C. Hsieh, K. Chiang, and I. Dhillon. Low rank modeling of signed networks. In *KDD*, pages 507–515. ACM, 2012.

[11] A. Jøsang and R. Ismail. The Beta reputation system. In *Proc. of the 15th Bled Electronic Commerce Conference*, volume 160, Bled, Slovenia, June 2002.

[12] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The Eigentrust algorithm for reputation management in p2p networks. In *WWW*, pages 640–651. ACM, 2003.

[13] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD*, pages 426–434. ACM, 2008.

[14] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[15] U. Kuter and J. Golbeck. Sunny: A new algorithm for trust inference in social networks using probabilistic confidence models. In *AAAI*, pages 1377–1382, 2007.

[16] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. In *WWW*, pages 641–650. ACM, 2010.

[17] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *KDD*, pages 177–187. ACM, 2005.

[18] G. Liu, Y. Wang, and M. Orgun. Optimal social trust path selection in complex social networks. In *AAAI*, pages 1391–1398, 2010.

[19] G. Liu, Y. Wang, and M. Orgun. Trust transitivity in complex social networks. In *AAAI*, pages 1222–1229, 2011.

[20] X. Liu, A. Datta, K. Rzadca, and E. Lim. Stereotrust: a group based personalized trust model. In *CIKM*, pages 7–16. ACM, 2009.

[21] H. Ma, M. Lyu, and I. King. Learning to recommend with trust and distrust relationships. In *RecSys*, pages 189–196. ACM, 2009.

[22] P. Massa and P. Avesani. Controversial users demand local trust metrics: An experimental study on epinions. com community. In *AAAI*, pages 121–126, 2005.

[23] A. Mishra and A. Bhattacharya. Finding the bias and prestige of nodes in networks based on trust scores. In *WWW*, pages 567–576. ACM, 2011.

[24] V. Nguyen, E. Lim, J. Jiang, and A. Sun. To trust or not to trust? predicting online trusts using trust antecedent framework. In *ICDM*, pages 896–901. IEEE, 2009.

[25] M. Richardson, R. Agrawal, and P. Domingos. Trust management for the Semantic Web. In *ISWC*, pages 351–368. Springer, 2003.

[26] J. Sabater and C. Sierra. Reputation and social network analysis in multi-agent systems. In *AAMAS*, pages 475–482. ACM, 2002.

[27] D. Sirdeshmukh, J. Singh, and B. Sabol. Consumer trust, value, and loyalty in relational exchanges. *The Journal of Marketing*, pages 15–37, 2002.

[28] J. Tang, H. Gao, and H. Liu. mTrust: discerning multi-faceted trust in a connected world. In *WSDM*, pages 93–102. ACM, 2012.

[29] J. Tang, H. Liu, H. Gao, and A. Das Sarmas. etrust: understanding trust evolution in an online world. In *KDD*, pages 253–261. ACM, 2012.

[30] A. Tversky and D. Kahneman. Judgment under uncertainty: Heuristics and biases. *science*, 185(4157):1124–1131, 1974.

[31] G. Wang and J. Wu. Multi-dimensional evidence-based trust management with multi-trusted paths. *Future Generation Computer Systems*, 27(5):529–538, 2011.

[32] Y. Wang and M. P. Singh. Trust representation and aggregation in a distributed agent system. In *AAAI*, pages 1425–1430, 2006.

[33] Y. Wang and M. P. Singh. Formal trust model for multiagent systems. In *IJCAI*, pages 1551–1556, 2007.

[34] D. Watts and S. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.

[35] R. Xiang, J. Neville, and M. Rogati. Modeling relationship strength in online social networks. In *WWW*, pages 981–990. ACM, 2010.

[36] L. Xiong and L. Liu. Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Transactions on Knowledge and Data Engineering*, 16(7):843–857, 2004.

[37] Y. Yao, H. Tong, F. Xu, and J. Lu. Subgraph extraction for trust inference in social networks. In *ASONAM*, pages 163–170, 2012.

[38] Y. Yao, H. Tong, X. Yan, F. Xu, and J. Lu. Matrust: An effective multi-aspect trust inference model. *arXiv preprint arXiv:1211.2041*, 2012.

[39] C. Ziegler and G. Lausen. Propagation models for trust and distrust in social networks. *Information Systems Frontiers*, 7(4):337–358, 2005.