Trust Relationship Prediction in Alibaba E-Commerce Platform

Yukuo Cen, Jing Zhang⋆, Gaofei Wang, Yujie Qian, Chuizheng Meng, Zonghong Dai, Hongxia Yang, Jie Tang⋆, *Senior Member, IEEE*

Abstract—This paper introduces how to infer trust relationships from billion-scale networked data to benefit Alibaba E-Commerce business. To effectively leverage the network correlations between labeled and unlabeled relationships to predict trust relationships, we formalize trust into multiple types and propose a graphical model to incorporate type-based dyadic and triadic correlations, namely eTrust. We also present a fast learning algorithm in order to handle billion-scale networks. Systematically, we evaluate the proposed methods on four different genres of datasets with labeled trust relationships: Alibaba, Epinions, Ciao and Advogato. Experimental results show that the proposed methods achieve significantly better performance than several comparison methods (+1.7-32.3% by accuracy; $p << 0.01$, with t-test). Most importantly, when handling the real large networked data with over 1,200,000,000 edges (Ali-large), our method achieves 2,000× speedup to infer trust relationships, comparing with the traditional graph learning algorithms. Finally, we have applied the inferred trust relationships to Alibaba E-commerce platform: Taobao, and achieved 2.75% improvement on gross merchandise volume (GMV).

Index Terms—Social network; Trust relationship prediction

1 Introduction

E-Commerce platform has led to a fundamental change in the way that businesses interact with their customers. Almost all the famous platforms, such as Taobao[1] and Amazon[2], try to attract new customers or keep existing customers by developing sophisticated strategies to recommend products. Traditional recommendations usually use content-based, collaborative filtering-based or hybrid methods. All these methods essentially categorize users/products into different groups and make recommendations based on the grouping information. However, a recent survey shows that 84% consumers' purchase behaviors are strongly influenced by friends' behaviors or friends' recommendations[3]. Leveraging the trust relationships between customers can significantly help E-Commerce. This actually has been demonstrated by social recommendation [11], [21], [31], [42], which suggests that the recommendation performance can be significantly improved with trust relationships. However, social recommendation does not present successful applications in industry. For example, according to IBM's Black Friday report [1], social networks including Facebook, Twitter and YouTube only contribute 0.34% of all online sales on Black Friday. One of the big challenges for applying social recommendation to many E-Commerce platforms is that most of the trust relationships are unavailable. Online social networks such as Facebook and Twitter record many different types of social relationships, but not all relationships are trustful. Trust relationships often hide in the large number of online social relationships or sometimes are missing in some networks, e.g., family relationship may not exist in a professional social network. Thus, questions that arise are: can we leverage users' behavior log to infer trust relationships between users? How can the inferred trust relationships finally help product recommendations in E-Commerce system?

In this paper, we aim to systematically study the problem on Taobao, the E-Commerce platform of Alibaba. Taobao has more than 500,000,000 users and is one of the largest E-Commerce platforms in the world — merely on 11/11/2017, the sales within 24 hours reach 25 billion US dollar. Specifically, we target at inferring trustful relationships between users in Taobao. Figure 1 shows an example to illustrate the problem that we are dealing with. The user behavior log we have collected to study the problem consists of a large number of user behaviors such as purchase history, mobile records, GPS information, etc. We have some relationships annotated with different types of trust or distrust by users or annotators. The goal is to infer all the other trust and distrust relationships. One intuitive way is to train a supervised

- *Yukuo Cen and Chuizheng Meng are with the Department of Computer Science and Technology,Tsinghua University, Beijing, China, 100084. E-mail:cyk18@mails.tsinghua.edu.cn, mengcz13@mails.tsinghua.edu.cn*
- *Jing Zhang is with Information School, Renmin University of China, Beijing, China, 100872. E-mail:zhang-jing@ruc.edu.cn, corresponding author*
- *Gaofei Wang, Zhonghong Dai and Hongxia Yang are with Alibaba Group, Beijing, China, 100026. E-mail:allen.wang@alibaba-inc.com, qiaoyin.dzh@taobao.com, yang.yhx@alibaba-inc.com*
- *Yujie Qian is with the MIT Computer Science and Artificial Intelligence Laboratory, 32 Vassar Street, Cambridge, MA 02139. E-mail:yujieq@mit.edu*
- *Jie Tang is with the Department of Computer Science and Technology,Tsinghua University, and Tsinghua National Laboratory for Information Science and Technology (TNList), Beijing, China, 100084. E-mail:jietang@tsinghua.edu.cn, corresponding author*

1. http://www.taobao.com, Alibaba E-Commerce platform
2. http://www.amazon.com
3. http://www.nielsen.com/us/en/insights/news/2013/under-the-influence-consumer-trust-in-advertising.html

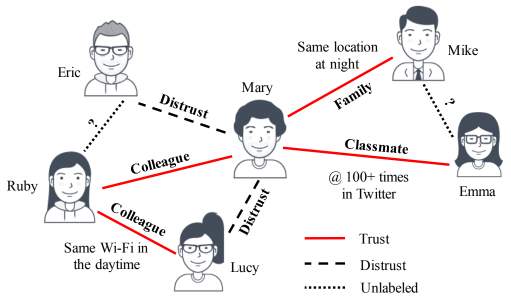


Fig. 1. Illustration of trust relationship prediction in social networks.

model by viewing each relationship as an instance. For example, in Figure 1, Mary and Mike always stay at the same location at night, which implies that they might be family relationship. However, on the Alibaba E-Commerce platform, we found that the real challenges lie in:

- **Limited labeled relationships.** In most cases, we only have a small number of labeled relationships, but all the other relationships are unlabeled. It is necessary to design a principled framework to learn with both limited labeled and large unlabeled relationships.
- **Billion-scale network.** The scale of the networked data is big with billions of nodes and edges. How to scale up the algorithm to handle such billion-scale data is a challenging issue.
- **Validation.** It is infeasible to directly evaluate the performance of inferred trust relationships on such a big network.

This paper investigates how to effectively leverage network correlations to infer trust relationships when a large number of neighboring relationships are unlabeled. To address the challenge of limited labeled relationships, semi-supervised learning such as label propagation [43] could help, by iteratively propagating trust scores from labeled relationships to unlabeled relationships. However, direct propagation is not always effective, due to the complex semantics of trust relationships. For example in Figure 1, Mary may not know Ruby's colleague Lucy, let alone trusts Lucy. In summary, the contributions of the work can be summarized as:

- We categorize trust into fine-grained types, and investigate potential useful correlations. We have discovered several intriguing correlation patterns (type-based dyadic and triadic correlations).
- Incorporating the discovered type-based dyadic and triadic correlations into a factor graph model, we propose eTrust, which significantly improves the inferring accuracy (+1.7-32.3%) against the state-of-the-art methods.
- To scale up eTrust to handle billion-scale networks, we propose eTrust-s, which achieves $> 2000\times$ speed-up than eTrust, with comparable accuracy performance.
- We have applied the inferred trust relationships to two Alibaba online products—Taobao product search and Taobao product discovery. Online A/B test shows that the proposed method can greatly improve the gross merchandise volume (GMV) by +2.75%.

2 Existing Methods Exploration

In this section, we explore possible solutions and analyze their limitations.

Unsupervised Methods. Unsupervised methods usually leverage network structures to estimate a trust score between two users. State-of-the-art methods such as Trust Propagation (TP) [10] and TidalTrust (TT) [8] propagate trust scores along edges in a network. A number of unsupervised link prediction approaches such as Common Neighbors (CN), Adamic/Adar (AA) and Jaccards Coefficient (JC) [17] can also be used to estimate trust scores. These methods avoid iterative propagation, but only aggregate trust scores from direct neighbors. Recent graph embedding methods such as DeepWalk (DW) [25] can also be used to first learn an embedding vector for each user and then calculate the trust score as dot product of two vectors.

Unsupervised methods highly depend on the network comprised of existing trust relationships. However, on the Alibaba platform, we only collected a small number of labeled relationships, and the connectivity of the formed network is quite poor (Graph density is only 3.13E-6), which makes it difficult to propagate trust scores.

Supervised Methods. Supervised methods such as logistic regression (LR) [7] train a multi-label classifier to predict the label of a relationship based on the features extracted from heterogeneous user behaviors. Related work of trust relationship prediction include [13], [19], [24]. Supervised methods for link prediction such as local Markov Random Field [38], supervised random walk [4] and a framework that incorporates all the unsupervised metrics as features into a supervised model [18], can also be leveraged to predict trust. Alibaba has advantaged extra sources to extract rich attribute features, in addition to the network comprised of labeled relationships. Thus compared with unsupervised methods, supervised methods can be applied to users with few labeled relationships. A few works extract features from network structure and incorporate them into a supervised model. For example, to predict positive and negative relationships, Leskovec et al. define triadic correlation features according to the theories of social balance and social status [16], [15]. Unfortunately, the sparsity of the labeled relationships on Alibaba's dataset impacts the effect of the correlation features, which are only extracted from the labeled network.

Semi-supervised Methods. Semi-supervised methods can incorporate the unlabeled relationships to predict trust. For example, Label Propagation (LP) [43] begins from the initially collected labels, and propagates trust scores along the network. LP is originally proposed to infer the labels of nodes. To adopt LP to predict trust, we can transform each relationship into a node, and build an edge between two nodes if their corresponding relationships

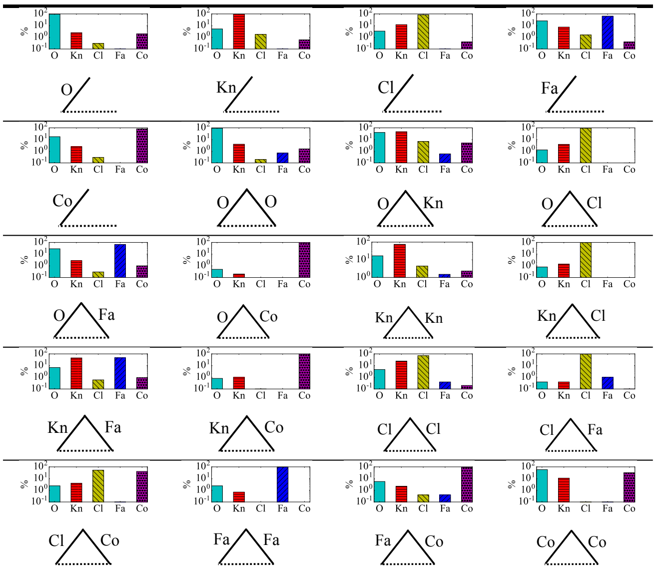
TABLE 1
Summarization of related methods. Notation t_{ij} is the trust score between user v_i and v_j, $\mathcal{V}_i$ denotes the neighbors of v_i, f is a real-valued function to estimate the trust score of a relationship e_i and y_i is the label of e_i.

Method	Equation	Explanation
CN [17]	$t_{ij} = \lvert\mathcal{V}_i \cap \mathcal{V}_j\rvert$	The number of common neighbors between v_i and v_j.
AA [3]	$t_{ij} = \sum_{v_k \in \mathcal{V}_i \cap \mathcal{V}_j} \frac{1}{\log(d(v_k)+1)}$	Weight rarer neighbors more heavily by $\frac{1}{\log(d(v_k)+1)}$ and $d(v_k)$ is the degree of v_k.
JC [27]	$t_{ij} = \lvert\mathcal{V}_i \cap \mathcal{V}_j\rvert / \lvert\mathcal{V}_i \cup \mathcal{V}_j\rvert$	The Jaccard similarity between v_i and v_j.
DW [25]	$t_{ij} =< \vec{v_i}, \vec{v_j} >$	$\vec{v}$ is learned by DeepWalk based on network structures.
TP [10]	$\widetilde{T} = (\alpha_1 T + \alpha_2 T^T T + \alpha_3 T^T + \alpha_4 T T^T)^k$	T is the initial trust matrix and $\widetilde{T}$ is the trust matrix after the k-th propagation.
TT [8]	$t_{ij} = \sum_{k \in \mathcal{V}_i \wedge t_{ik} > r} t_{ik} t_{kj} / \sum_{k \in \mathcal{V}_i \wedge t_{ik} > r} t_{ik}$	r is a threshold value to select the most trusted neighbors.
LR [7]	$f(\mathbf{x}_i) = 1/(1 + \exp\{-\alpha^T \mathbf{x}_i\})$	$\mathbf{x}_i$ is the attribute vector of e_i and α is the corresponding weighting vector.
LP [43]	$E(f) = \frac{1}{2} \sum_{i,j} \omega_{ij} (f_i - f_j)^2$	ω_{ij} indicates the similarity score of e_i and e_j, and E is the energy function.

share a common user. The objective function is defined as $E(f) = \frac{1}{2}\sum_{i,j}\omega_{ij}(f_i - f_j)^2$, where ω_{ij} is the similarity score of two relationships e_i and e_j, which is calculated as the dot product of their attribute vectors $\mathbf{x}_i$ and $\mathbf{x}_j$. Notation f is a real-valued function $f : E \rightarrow \mathbf{R}$. The initial value of f_i equals 1 if e_i is trustful, $f_i = -1$ if e_i is distrustful, and $f_i = 0$ if e_i is unlabeled. Essentially, the value of f at each unlabeled relationship is the average of f at neighboring relationships, and f at each labeled relationship is constrained to take its initial value after each iteration. Label Spreading (LS) [41] smooths the value of f by its initial value at each iteration. LP and LS both assume that the classified labels should not change too much between nearby nodes. But the transitivity of trust may not be established under some scenarios when multi-typed semantics of trust is considered.

Recently proposed semi-supervised graph convolutional/attention models [12], [37] represent a node's embedding by performing convolutional operations on its neighbors' embeddings and predict nodes' labels based on their embeddings. However, they suffer from the same transitivity assumption as LP/LS and the input of a large network prevents efficient mini-batch training. Tang et al. [36], [33] and Lou et al. [20] formalized the social balance theory into triadic correlation factors and incorporated them into factor graphical models to predict links. However, they did not discuss the situations when the relationships have multiple types and also thoroughly ignored the dyadic correlations between relationships. Table 1 summarizes the details of the major related methods.

TABLE 2
Statistics of dyadic and triadic correlation patterns (Y-axis: log scale).



3 Dyadic/Triadic Correlations

In this section, we present the dyadic and triadic correlation patterns discovered from the collected Alibaba dataset.

The semantics of trust is complex, with different interpretations in different contexts [2]. For example, in an E-Commerce platform, people may trust their acquaintances to buy their recommending products, while in Advogato, an online social networking site dedicated to free software development, people may only trust technical experts. In this paper, we define trust relationships according to the social theory of Dunbar number [22]. Specifically, people typically have a few ultra-close family members, followed by less cozy companions including classmates and colleagues, and other acquaintances at more distance or friends except the former three types. Formally, we categorize the trust semantics of a relationship into four types, i.e., $\mathcal{T}$={*Fa*, *Cl*, *Co*, *Kn*}, where *Fa* denotes family, *Cl* is classmate, *Co* is colleague and *Kn* means other known relationships or friends except the former three types. We use notation *O* to represent the type of distrust relationships. This definition also refers to the statements that the social ties which we have with other people in social networks usually include our families, colleagues, classmates and friends [28] and those social ties build trust between people [26]. Following

TABLE 3
Notations.

SYMBOL	DESCRIPTION
$\mathcal{T}$={*Fa*, *Cl*, *Co*, *Kn*}	The type of trust relationships. *Fa*, *Cl*, *Co* and *Kn* are family, classmate, colleague and friend respectively.
O	The type of distrust relationships
E^L and E^U	E^L and E^U are labeled and unlabeled relationships respectively.
$Y^L$and Y^U	Y^L are the labels of E^L and Y^U are the labels of E^U to be inferred.
$\mathbf{X}$	The relationship attribute matrix.
$\wedge_{ij}$	Two neighboring relationships e_i and e_j
$\triangle_{ijk}$	Three neighboring relationships e_i, e_j and e_k in a triadic structure.

this definition, the problem of trust relationship prediction is simplified as predicting the four types of relationships between users.

Now we introduce how to derive a social network from the E-Commerce data, as many E-Commerce platforms do not really have a social network. We construct the social network by considering two aspects, interaction and homophily. Interactions between users include the communication times by mobile, the comment and chat times by social tools and so on. Homophily [14], [29] is represented as the similarity between user attributes or behaviors, such as the number of same visited locations, the number of same purchased products and so on. We build a relationship between two users if they have more than τ interactions or if the homophily is larger than threshold μ, where τ and μ are heuristic parameters that can be tuned according to different datasets. We set the thresholds to reserve the most possible candidate trust relationships and also avoid the explosion of the relationships in the network. In this way, we can derive a network $G = (V, E)$, where V is the set of users and $E \subset V \times V$ is the set of relationships. Some trust relationships can be easily identified based on the types of interactions, and some relationships can be annotated manually according to their attribute or behavioral patterns. For example, if two users always stay at the same location at night, it is highly probable that they are family, while they are quite likely to be colleagues if they always stay at the same location in working hours. Thus we could have a small portion of labeled relationships. To summarize, the input of our problem can be defined as:

Definition 1: **Partially labeled attribute augmented network**: A partially labeled attribute augmented network is a network with the relationships partially labeled. We denote the network as $G = (V, E^L, E^U, Y^L, \mathbf{X})$, where E^L is the set of labeled relationships with Y^L as the corresponding labels, and E^U is the set of unlabeled relationships satisfying $E^L \cup E^U = E$ and $E^L \bigcap E^U = \varnothing$. Notation $\mathbf{X}$ denotes an attribute matrix associated with the relationships in E, where each row corresponds to a relationship e_i, each column represents an attribute feature, and an element x_{id} denotes the value of the d-th attribute of e_i.

Based on the labeled relationships in the derived network G, we conduct an analysis on network correlations between neighboring relationships and present the results in Table 2, where in each dyadic or triadic structure, the solid edges represent the observed neighboring relationships with labels, and the dashed edges are the relationships to be investigated. For each dyadic or triadic structure, we present the probability of the relationship to be investigated belonging to one of the four trust types in $\mathcal{T}$ or the distrust type in the bar graph. We consider the triangle structures the most in our network, as the influence weakens and the computational burden increases when the length of the relationship path increases. The experiments also verify that the dyadic and triadic structures can obtain best accuracy and efficiency performance when predicting trust.

We observe that in most triadic structures, the type of a relationship is very likely to be the same with one of the other two relationships. An exception occurs when two neighboring relationships are both colleagues (*Co*), while the third one is most likely to be a distrust relationship. This may due to the reason of frequent job-hopping. We also find that in dyadic structures, two neighboring relationships have a high probability to be the same type. This is because different types of labeled relationships were collected separately from different groups of users, and may cause the clustering of the relationships with same labels. We also conduct the same analysis on Epinions and Ciao — two product review sites that allow users to create explicit trust relationships with others. There are only binary types of relationships in the two datasets. i.e., trust and distrust relationships. We find that the types of two neighboring relationships do not have close correlation (In a dyadic structure, when one relationship is trustful, the probability of the other relationship being trustful is only around 50%). However, the transitivity assumption of trust can be strongly verified (i.e., it is very likely to deduce A trusting C when A trusts B and B also trusts C).

In summary, in different networks, we can not unitedly assume the neighboring relationships tend to be the same types. Besides, under the scenarios when multi-typed semantics of trust is considered, we cannot make the assumption of transitivity of trust. In another word, different network correlation factors exert different effect on predicting trust in different networks. We automatically learn the contributions of different factors by model itself instead of making certain assumptions. The major notations used in the paper are summarized in Table 3.

4 The Proposed Models

Unsupervised methods suffer from sparse connections among labeled relationships. Supervised methods capture rich attribute features, but ignore network structures due to the same reason. Semi-supervised methods leverage both labeled and unlabeled relationships, but the transitivity assumption of trust is not always established when trust is multi-typed. Considering all the the limitations, we propose two models to incorporate the discovered dyadic and triadic correlations between labeled or unlabeled relationships to

Algorithm 1: eTrust

Input: Network $G = (V, E^L, E^U, Y^L, \mathbf{X})$, learning rate η
Output: Learned parameters $\theta = \{\alpha, \beta, \gamma\}$

Initialize θ ;
repeat
  Calculate each $p(y_i)$, $p(y_i, y_j)$ and $p(y_i, y_j, y_k)$ by LBP;
  Calculate each $p(y_i)$, $p(y_i, y_j)$ and $p(y_i, y_j, y_k)$ conditioned on the observed labels by LBP;
  Calculate the expectations by Eq. (9) ;
  Calculate the gradients of θ according to Eq. (8);
  Update parameters θ with the learning rate η:

$$\theta_{new} = \theta_{old} - \eta \nabla_\theta$$

until *Convergence*;

predict trust on Alibaba E-Commerce platform, namely eTrust and eTrust-fast (abbreviated as eTrust-s).

4.1 Problem Formulation

Our problem can be formalized as:

Problem 1: **Trust relationship prediction on Billion-scale Network.** Given a partially labeled attribute network $G = (V, E^L, E^U, Y^L, \mathbf{X})$, the objective of trust relationship prediction is to learn a predictive function

$$f : G = (V, E^L, E^U, Y^L, \mathbf{X}) \rightarrow Y \tag{1}$$

that can be used to predict the type y_i for each unlabeled relationship $e_i \in E^U$ where y_i is a value in $\mathcal{T} \cup \{O\}$ with O represents distrust. Notation $Y = \{y_i\}_{i=1}^{|E|}$ represents the label set of all the relationships.

It is worth noting that as the prediction is performed on a billion-scale network, it is infeasible to first learn a model from the huge network and then apply it to the huge network again. Learning and prediction should be done in a unified and efficient way. Finally, the relationships that are predicted into one type of $\mathcal{T}$ are collected as the discovered trust relationships.

4.2 eTrust

The basic idea of eTrust is to incorporate the type-based dyadic and triadic correlations as factors into a factor graphical model. We transform the original node-oriented network into a relation-oriented network (i.e., line graph) by formalizing each relationship as a node, and adding an edge between two nodes if their corresponding relationships in the original network share a common user. Given a line graph G, we factorize the joint distribution over the label set Y of all the relationships in G into a product of factors, with each factor representing a function of a set of variables, where a variable indicates the attribute vector or the label of a relationship in the graph:

$$p(Y|G) = \frac{1}{Z} \prod_{e_i} f(y_i, \mathbf{x}_i) \prod_{\wedge_{ij}} g(y_i, y_j) \prod_{\triangle_{ijk}} h(y_i, y_j, y_k), \tag{2}$$

where $f(y_i, \mathbf{x}_i)$ is the factor function corresponding to a relationship e_i, which represents the correlation between e_i's label y_i and the attribute vector $\mathbf{x}_i$; Notation $g(y_i, y_j)$ denotes the factor function corresponding to two neighboring relationships e_i and e_j[4], which represents the correlation between the two relationships. Notation $h(y_i, y_j, y_k)$ denotes the factor function corresponding to three relationships e_i, e_j and e_k in a triadic structure[5], which represents the correlation between the three relationships. Notation Z denotes the global normalization term that adds up the products of the factor functions over all possible configurations of the relationships' labels, i.e., $Z = \sum_Y \prod_{e_i} f(y_i, \mathbf{x}_i) \prod_{\wedge_{ij}} g(y_i, y_j) \prod_{\triangle_{ijk}} h(y_i, y_j, y_k)$. We introduce the details of the factor functions as follows.

Factor Functions. The attribute factor function $f(y_i, \mathbf{x}_i)$ is related to both the label and the attribute vector:

$$\begin{aligned} f(y_i, \mathbf{x}_i) &= \exp\{\alpha^T \phi(y_i, \mathbf{x}_i)\}, \qquad (3) \\ \phi(y_i, \mathbf{x}_i) &= (\mathbb{1}_{y_i=Fa}\mathbf{x}_i, \mathbb{1}_{y_i=Cl}\mathbf{x}_i, \mathbb{1}_{y_i=Co}\mathbf{x}_i, \mathbb{1}_{y_i=Kn}\mathbf{x}_i, \mathbb{1}_{y_i=O}\mathbf{x}_i)^T, \end{aligned}$$

where $\mathbf{x}_i$ is a D-dimension attribute vector, $\mathbb{1}_{y_i=Fa}$ is the indicator function with value as 1 when $y_i = Fa$ and 0 otherwise. Then $\mathbb{1}_{y_i=Fa}\mathbf{x}_i$ equals $\mathbf{x}_i$ when $\mathbb{1}_{y_i=Fa} = 1$ and a zero vector $\mathbf{0}$ otherwise. The vector $\phi(y_i, \mathbf{x}_i)$ and the weighting vector α are both $5 \times D$-dimension vector. For example, when $y_i = Fa$, $\phi(y_i, \mathbf{x}_i) = (\mathbf{x}_i, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0})$. The attribute vector $\mathbf{x}_i$ contains the features extracted from the aspects of interactions, homophily [14], [29] and social influence [5], [9] to indicate whether two users trust each other or not. Interactions denote the frequency of interactions between users such as the communication times by mobile. Homophily reflects the similarity of two users such as age, gender or the number of the same products purchased. Social influence occurs when one's behaviors or opinions are affected by others, which is simply quantified by the frequency that one user performs a same behavior after another user.

The dyadic factor function $g(y_i, y_j)$ and the triadic factor function $h(y_i, y_j, y_k)$ are respectively defined as:

$$\begin{aligned} g(y_i, y_j) &= \exp\{\beta^T \psi(y_i, y_j)\}, \qquad (4) \\ h(y_i, y_j, y_k) &= \exp\{\gamma^T \zeta(y_i, y_j, y_k)\}, \end{aligned}$$

where β and γ are the weighting vectors, and ψ and ζ are vectors of dyadic and triadic correlation feature functions respectively, with each feature function in them defined as:

$$\begin{aligned} \psi_{a,b}(y_i, y_j) &= \begin{cases} 1 & y_i = a, y_j = b; \\ 0 & \text{otherwise,} \end{cases} \qquad (5) \\ \zeta_{a,b,c}(y_i, y_j, y_k) &= \begin{cases} 1 & y_i = a, y_j = b, y_k = c; \\ 0 & \text{otherwise,} \end{cases} \end{aligned}$$

where a, b and c represent specific relationship types and they can be any trust type in $\mathcal{T} = \{Fa, Cl, Co, Kn\}$ or the

4. We represent two neighboring relationships as $\wedge_{ij}$.
5. We represent three relationships in a triadic structure as $\triangle_{ijk}$.

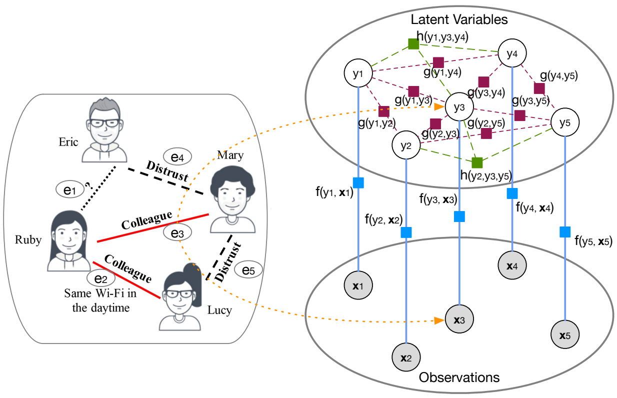


Fig. 2. Illustration of eTrust. Each relationship between two users is corresponding to a feature vector **x** and a label variable y. Attribute factor function f is defined on **x** and y; dyadic correlation function g is defined between the label variables of two neighboring relationships; tradic correlation function h is defined between the label variables of three neighboring relationships in a triadic structure.

distrust type. So $\psi = (\psi_{Fa,Fa}, \psi_{Fa,Cl}, \psi_{Fa,Co}, \psi_{Fa,Kn}, \cdots)$ is actually a one-hot vector with only one dimension as 1. For example, when $y_i = Fa$ and $y_j = Co$, $\psi(y_i, y_j) = (0, 0, 1, 0, \cdots)$. Vector ζ can be similarly explained. The order of a, b, and c that are assigned to y_i, y_j and y_k does not impact the results, because the fine-grained trust semantics *Fa*, *Cl*, *Co* and *Kn* indicate undirected characteristics of trust relationships. Based on the 4 defined trust types and 1 distrust type, there are $\binom{5}{1} + \binom{5}{2} = 15$ dyadic correlation features and $\binom{5}{1} + 2 * \binom{5}{2} + \binom{5}{3} = 35$ triadic correlation features in total, where different relationships in a dyadic or triadic structure can be assigned a same type. We take part of the social network in Figure 1 to illustrate how the factor graphic model—eTrust is constructed in Figure 2.

Model Learning. We learn model parameters by maximizing the log-likelihood function of all the labeled relationships:

$$\mathcal{O}(\alpha, \beta, \gamma) = \log P(Y^L|G) = \log \sum_{Y|Y^L} P(Y|G) \tag{6}$$

By plugging Eq. (2) into the above equation, we obtain

$$\begin{aligned} \mathcal{O}(\alpha, \beta, \gamma) &= \log \sum_{Y|Y^L} \exp[\sum_{e_i} \alpha^T \phi(y_i, \mathbf{x}_i) \\ &+ \sum_{\wedge_{ij}} \beta^T \psi(y_i, y_j) + \sum_{\triangle_{ijk}} \gamma^T \zeta(y_i, y_j, y_k)] \\ &- \log \sum_{Y} \exp[\sum_{e_i} \alpha^T \phi(y_i, \mathbf{x}_i) \\ &+ \sum_{\wedge_{ij}} \beta^T \psi(y_i, y_j) + \sum_{\triangle_{ijk}} \gamma^T \zeta(y_i, y_j, y_k)] \end{aligned} \tag{7}$$

We use gradient descent algorithm to learn the parameters $\theta = \{\alpha, \beta, \gamma\}$. The gradient of γ is as follows (Other gradients can be calculated in the same way):

$$\begin{aligned} \frac{\partial \mathcal{O}}{\partial \gamma_{a,b,c}} &= \mathbb{E}_{p(Y|Y^L)} \sum_{\triangle_{ijk}} \zeta_{a,b,c}(y_i, y_j, y_k) \\ &- \mathbb{E}_{p(Y)} \sum_{\triangle_{ijk}} \zeta_{a,b,c}(y_i, y_j, y_k), \end{aligned} \tag{8}$$

where $Y|Y^L$ denotes a labeling configuration Y of all the relationships given the observed labels Y^L and $p(Y|Y^L)$ is the corresponding probability. Notation $\mathbb{E}_{p(Y|Y^L)} \sum_{\triangle_{ijk}} \zeta_{a,b,c}(y_i, y_j, y_k)$ denotes the expectation of the summation of a triadic correlation feature $\zeta_{a,b,c}(y_i, y_j, y_k)$ given the label distribution over all the unlabeled relationships conditioned on the labeled ones, and $\mathbb{E}_{p(Y)} \sum_{\triangle_{ijk}} \zeta_{a,b,c}(y_i, y_j, y_k)$ is the expectation of the same feature given the label distribution over all the relationships. We can rewrite the expectation $\mathbb{E}_{p(Y)} \sum_{\triangle_{ijk}} \zeta_{a,b,c}(y_i, y_j, y_k)$ as:

$$\begin{aligned} \mathbb{E}_{p(Y)} \sum_{\triangle_{ijk}} \zeta_{a,b,c}(y_i, y_j, y_k) &= \sum_{Y} p(Y) \sum_{\triangle_{ijk}} \zeta_{a,b,c}(y_i, y_j, y_k) \\ &= \sum_{\triangle_{ijk}} \sum_{y_i, y_j, y_k} \zeta_{a,b,c}(y_i, y_j, y_k) p(y_i, y_j, y_k) \end{aligned} \tag{9}$$

Eq. (9) indicates that, to calculate the expectation, we need to know the marginal probability $p(y_i, y_j, y_k)$. Loopy Belief Propagation (LBP) [23] is one popular approximate algorithm to calculate the marginal probabilities in a graphical structure. The key idea is to define a sum-product update rule to pass messages between the factor nodes and the variable nodes in a factor graph and calculate the marginal probability of a variable node as the product of the messages passing to it. Algorithm 1 presents the algorithm details. At the beginning, we perform LBP to obtain all the demanded marginal probabilities $p(y_i)$, $p(y_i, y_j)$ and $p(y_i, y_j, y_k)$ (Line 3) and perform LBP again to obtain those marginal probabilities conditioned on the observed labels (Line 4). Then we calculate the expectations based on the marginal probabilities by Eq. (9) (Line 5). Finally we compute the gradients based on the expectations by Eq. (8) (Line 6) and update the parameters by the gradients (Line 7). Thus, LBP inference of marginal probabilities is the prerequisite of gradient update in each iteration.

Inferring Unlabeled Trust Relationships. Based on the learned parameters θ, we can predict the unlabeled trust relationships by finding a label configuration that maximizes the joint probability of both labeled and unlabeled relationships using LBP, i.e., $Y^* = \text{argmax}_{Y|Y^L} p(Y|G, \theta)$.

4.3 Fast Learning for eTrust

The proposed eTrust is inefficient when the input network is large, due to LBP process. LBP is an iterative process and needs to enumerate all possible label configurations for all the relationships, dyadic structures and triadic structures at each iteration. Thus, the time complexity of LBP process is proportional to $O(I_{LBP}(NT + MT^2 + WT^3))$, where I_{LBP} is the iterative times of LBP, N, M and W are the number of relationships, dyadic structures and triadic structures respectively, and T is the number of types in $\mathcal{T}$ plus the distrust type. By further considering the outermost iteration I, the whole time complexity of Algorithm 1 is proportional to $O(II_{LBP}(NT+MT^2+WT^3))$. We further propose eTrust-s, which is much more efficient than eTrust, but can incorporate the same factors. eTrust-s tries to avoid calculating the marginal probability $p(y_i, y_j)$ for a dyadic structure and $p(y_i, y_j, y_k)$ for a triadic structure. The key challenge is at each iteration, the labels of the neighboring relationships in those structures are all unknown, making the computation of the marginal probability of each relationship interdependent. To separate these, when calculating the marginal probability of a relationship's label y_i, we assume the labels of its neighboring relationships $Y_{\mathcal{N}_i}$ are known, where $\mathcal{N}_i$ denotes the neighbors of relationship e_i. The solution is to use the inferred labels of last iteration as their values at this iteration. Essentially, eTrust-s calculates $p(y_i|\hat{Y}_{\mathcal{N}_i})$ instead of $p(y_i, y_j)$ and $p(y_i, y_j, y_k)$, where the notations with ˆ are the inferred labels. Thus, the log-likelihood function of eTrust-s is defined as:

$$\begin{aligned}\mathcal{O}(\theta) &= \sum_{e_i}(\alpha^T\phi(y_i, \mathbf{x}_i) + \sum_{\wedge_{ij}}\beta^T\psi(y_i, \hat{y}_j) \\ &+ \sum_{\triangle_{ijk}}\gamma^T\zeta(y_i, \hat{y}_j, \hat{y}_k) - \log Z_i),\end{aligned} \tag{10}$$

where Z_i is a local normalization term over the value of y_i instead of a global normalization term over the values of all the relationships in eTrust, because given the labels of all the neighboring relationships $\hat{Y}_{\mathcal{N}_i}$, the marginal probability $p(y_i|\hat{Y}_{\mathcal{N}_i})$ of e_i is independent from other relationships. Specifically, Z_i is represented as:

$$\begin{aligned}Z_i &= \sum_{y_i}\exp\{\alpha^T\phi(y_i, \mathbf{x}_i)\} + \sum_{y_i}\exp\{\sum_{\wedge_{ij}}\beta^T\psi(y_i, \hat{y}_j)\} \\ &+ \sum_{y_i}\exp\{\sum_{\triangle_{ijk}}\gamma^T\zeta(y_i, \hat{y}_j, \hat{y}_k)\}.\end{aligned} \tag{11}$$

Then β is calculated using gradient descent method (Parameters α and γ are calculated in the same way):

$$\begin{aligned}\frac{\partial\mathcal{O}}{\partial\beta_{a,b}} &= \sum_{e_i}(\sum_{\wedge_{ij}}\psi_{a,b}(y_i, \hat{y}_j) \\ &- \sum_{y_i'}p(y_i'|\hat{Y}_{\mathcal{N}_i})\sum_{\wedge_{ij}}\psi(y_i', \hat{y}_j)),\end{aligned} \tag{12}$$

where $p(y_i|\hat{Y}_{\mathcal{N}_i})$ is instantiated as:

$$p(y_i|\hat{Y}_{\mathcal{N}_i}) = \frac{\exp\sum_{\wedge_{ij}}\beta^T\psi(y_i, \hat{y}_j)}{\sum_{y_i'}\exp\sum_{\wedge_{ij}}\beta^T\psi(y_i', \hat{y}_j)}. \tag{13}$$

Algorithm 2 presents the learning process of eTrust-s. We initialize β and γ randomly (Line 1) and initialize α as the optimized parameter when only considering attribute features in Eq. (3) (Line 2). We divide the labeled relationships into batches $\{E_k^L\}_{k=1}^B$ (Line 3) and iteratively process the randomly shuffled batches (Line 4, 5 and 6). For each selected batch, we infer the labels for all the unlabeled neighboring relationships of each labeled relationship (i.e., $\{y_j|y_j \in Y^U \cap e_j \in \mathcal{N}_i \cap e_i \in E_k^L\}$) by current model (Line 7), and calculate the correlation features defined in Eq. (5) for each labeled relationship given the inferred labels of its neighboring relationships $\hat{Y}_{\mathcal{N}_i}$ (Line 8). Essentially, we count the frequency of all kinds of correlation features for each labeled relationship. Finally, we update parameters θ on the current selected batch of labeled relationships E_k^L (Line 9). The time complexity of calculating correlation features in eTrust-s is only $O(N+M+W)$. Thus, the whole time complexity is reduced from $O(II_{LBP}(NT+MT^2+WT^3))$ in eTrust to $O(I(N+M+W))$ in eTrust-s. When $T=5$, eTrust-s can almost achieve 125× speed up. Additionally, as the iterative times of LBP—I_{LBP} is avoid and the outermost iteration I of eTrust-s is much smaller than that of eTrust (The experimental results show that eTrust-s can converge in at most 50 iterations, but eTrust usually needs hundreds of iterations to converge), the speedup of eTrust-s is far more than 125×. After model training, we get the labels of all the unlabeled relationships Y^U at the same time.

Algorithm 2: eTrust-s

Input: Network $G=(V, E^L, E^U, Y^L, \mathbf{X})$, learning rate η
Output: Learned parameters $\theta=\{\alpha, \beta, \gamma\}$
1 Initialize β and γ randomly;
2 Initialize α through optimizing Eq. (3);
3 Divide labeled relationships into batches $\{E_k^L\}_{k=1}^B$;
4 **for** *epoch l=1 to L* **do**
5   Shuffle batches $\{E_k^L\}_{k=1}^B$;
6   **foreach** $E_k^L \in E^L$ **do**
7   Infer $\{y_j|y_j \in Y^U \cap e_j \in \mathcal{N}_i \cap e_i \in E_k^L\}$ by the model with current parameters θ;
8   Calculate the correlation features (Eq. (5)) for each $e_i \in E_k^L$ given the inferred labels of its neighbors $\hat{Y}_{\mathcal{N}_i}$;
9   Calculate the gradients of θ and update the parameters by $\theta_{new}=\theta_{old}-\eta\nabla_\theta$;

Distributed Learning. By eTrust-s, the computation of a batch of data can be easily distributed. The algorithm contains two key operations for each batch. The first operation is, for each labeled relationship in a batch, we infer the labels of its neighboring relationships and then calculate its correlation features based on the inferred neighboring labels. To distribute this operation, we save the whole network in an adjacent matrix with each row indicating a relationship and all its neighboring relationships, and then distribute the rows of the adjacent matrix into different machine nodes. In this way, no information is lost, as the dyadic and triadic correlation features are formed by a relationship and its neighboring relationships, which are all kept in the same machine node. The second operation is to calculate the gradients of the model parameters based on the batch of data, which is independent among the labeled relationships, and can also be distributed easily. In real implementations, we distribute eTrust-s in ODPS, a large distributed computation platform of Alibaba inc..

TABLE 4
Dataset statistics. Notation $|V|$, $|E|$ and $|E^L|$ denote the number of users, relationships and labeled relationships respectively. Notation $D = \frac{2|E|}{|V|(|V|-1)}$ denotes graph density.

Dataset	$\|V\|$	$\|E\|$	$\|E^L\|$	D
Alibaba	86,550	202,624	59,613	5.41E-5
Ali-large	28,435,081	1,264,260,801	59,613	3.13E-6
Epinions	812	44,240	44,240	0.134
Ciao	1,014	66,326	66,326	0.129
Advogato	3,189	16,594	16,594	3.00E-3

TABLE 5
Efficiency performance(CPU time of model learning, s:seconds, m:minutes, h:hours).

Dataset	eTrust	eTrust-s	Speedup
Alibaba	16.70h	27.27s	2205×
Epinions	97.30m	1.49s	3918×
Ciao	72.60m	2.33s	1870×
Advogato	32.90m	0.90s	2193×

5 EXPERIMENT

5.1 Experimental Settings

5.1.1 Datasets

We mainly evaluate the proposed methods on the collected Alibaba's dataset.

Alibaba: According to the method about how to derive a network in Section 3, We collected 28,435,081 active users from Alibaba's platform and built 1,264,260,801 relationships among them. The dataset is named as **Ali-large**. Then we identify or annotate labels for 59,629 relationships, which include 9,580 *Fa*, 10,066 *Cl*, 10,166 *Co*, 10,031 *Kn* and 19,770 *O* types of relationships (Cf. Section 3 for details of how to derive the network).

In order to execute all the comparison methods in available time, we extract a small network **Alibaba** from Ali-large by expanding the neighbors of all the users that are included in the annotated 59,629 user pairs (i.e., relationships), and result in 86,550 users and 202,624 relationships among all these users. We extract the dataset as above to make the dyadic and triadic structures of the labeled relationships complete enough to verify the effects of dyadic/triadic correlations. **Alibaba** has the same amount of labeled relationships with Ali-large, except that the unlabeled relationships are reduced. We only learn and predict the trust relationships in Ali-large by the efficient method eTrust-s, and indirectly evaluate them by online real applications of Aibaba Group instead of direct evaluation in terms of precision, recall and F1.

To further verify the generalization of the proposed methods, we also evaluate on three other datasets:

Epinions [30]: Epinions is a product review site. Users can rate the products by 1 to 5 scores, and can also create trust relationships with others. Each rating contains user name, product name, product's category and rating score. The trust relationships in Epinions are directed, which indicates that A trusting B may not deduce B trusting A. Besides, we only collect binary labels for its relationships.

Ciao [30]: Ciao is also a product review site. Users can also rate products and create trust relationships with others. The relationships are also directed and the labels are binary.

Advogato[6]: Advogato is a community and social networking site of free software developers, where developers share their developing skills and raters will rate them with three trust levels—Apprentice, Journeyer and Master. Apparently, the relationships are directed.

Essentially, trust semantics is flexible and can be defined as binary or multiple, directed or undirected, depending on the real applications. In E-Commerce applications, the trust relationships are more like acquaintances, which are naturally undirected, and the semantics can be multiple types. But in the rating sites Epinions and Ciao and the expert site Advogato, trust relationships are more like "celebrating following" relationships and are directed. Such directed trust elationships can also be exemplified by the advisor-advisee relationships in academic networks [35]. To define correlation features upon the directed trust relationships, we only consider the directed triangle structure "A trusts B, B trusts C, and A trusts C", as transitivity is one of the basic properties of trust [32]. Then we ignore the directions and define the correlation features according to Eq. (5).

To define attribute features, since we only collected a network in Advogato, we define attribute features as the difference between degree, clustering coefficient, closeness or pagerank scores of two users, etc. We also extract attribute features from the rating log such as the number of same products rated by two users in Epinions and Ciao. On the dataset of Alibaba, we have rich user behavior data and can define multiple types of user interactions, homophily and social influence as attribute features. Since we only collected binary-typed relationships in Epinions and Ciao, we conduct binary classification on Epinions and Ciao (i.e., reduce eTrust and eTrust-s into binary classifiers), and conduct multiple classification on Advogato and Alibaba. Table 4 lists the statistics of the datasets, where Alibaba is partially labeled, and Epinions, Ciao and Advogato are completely labeled.

5.1.2 Comparison Methods

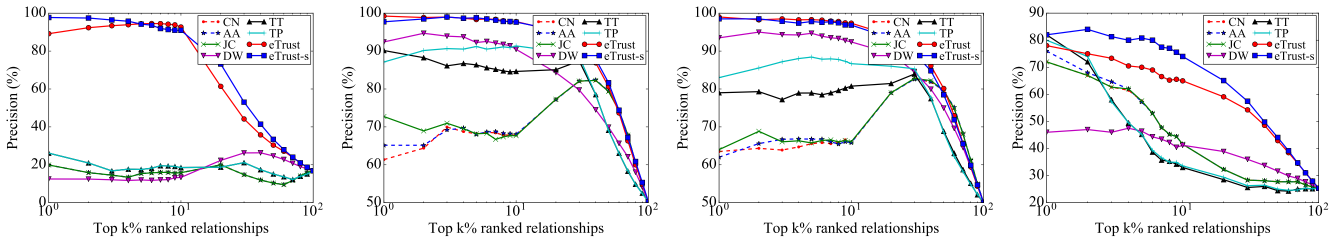
We compare unsupervised, supervised and semi-supervised methods. The labeled relationships in the datasets are divided into training and test set to learn and evaluate the comparison methods.

Unsupervised Methods: include Common Neighbors (CN) [17] which counts the common neighbors of two users, Adamic/Adar (AA) [3] which is similar to CN but weights a neighbor by its degree, Jaccard's Coefficient (JC) [27] which further considers the number of individual neighbors, DeepWalk (DW) [25] which calculates cosine similarity of two users based on their learned embedding vectors, Trust Propagation (TP) [10] and TidalTrust (TT) [8] which not only consider the number of direct neighbors but also the high-order neighbors to calculate the trust score between two users. When using the unsupervised

6. http://www.trustlet.org/wiki/Advogato_dataset.

TABLE 6
Prediction performance of the supervised and semi-supervised methods on four datasets (%).

Method	Alibaba				Epinions				Ciao				Advogato			
	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1	Acc.
LR	64.59	61.57	62.59	62.85	53.00	98.40	68.90	55.34	59.63	80.52	68.52	62.96	43.40	32.02	33.15	37.68
SVM	66.95	65.82	65.64	66.58	81.69	69.73	75.24	76.93	74.09	69.44	71.69	72.55	42.06	42.07	39.53	42.10
RF	70.95	69.90	70.32	71.04	80.08	76.53	78.26	78.63	73.37	69.32	71.29	72.05	54.12	53.92	53.99	53.90
LP	91.90	86.49	88.81	77.16	77.97	74.97	76.44	76.77	71.72	69.60	70.64	71.05	70.48	29.68	41.51	33.26
LS	90.80	86.94	88.51	76.69	71.62	82.30	76.59	74.71	67.59	76.56	71.80	69.89	73.64	22.69	33.56	28.98
eTrust	86.55	**91.73**	89.01	87.37	81.62	**78.55**	**80.06**	80.33	79.66	**80.29**	**79.98**	**79.88**	55.02	57.32	56.13	59.36
eTrust-s	**92.00**	90.56	**91.21**	**90.73**	**85.50**	74.89	79.84	**80.99**	**83.67**	69.92	76.18	78.12	**58.66**	**58.22**	**58.41**	**61.28**



(a) Alibaba (b) Epinions (c) Ciao (d) Advogato

Fig. 3. Precision at top k% of the unsupervised methods and our methods. X-axis: k (log scale); Y-axis: Precision.

methods, we cast our task as a ranking problem and evaluate each trust type separately. Specifically, for each trust type, we build a network including all the labeled relationships of this type in the training set, based on which we estimate a score by one of the above methods for each pair of users in the test set and rank them based on their scores.

Supervised methods: include Logistic Regression (LR), Support Vector Machine (SVM) and Random Forest (RF). They all define the same attribute features as the proposed methods. When using the supervised methods, we cast our task as a classification problem. Specifically, for both the labeled relationships in training data and test set, we extract the attribute features from the aspects of interactions, homophily and social influence the same as the proposed methods. Then we train a supervised model on the training data and predict a label for each relationship in the test set.

Semi-supervised methods: include Label Propagation (LP) [43] which propagates labels according to relationships' proximity, and Label Spreading (LS) [41] which is similar to LP but uses initial values to smooth the inferred values at each iteration. We build a line graph for each trust type, propagate the trust scores from the labeled relationships to the unlabeled relationships in the network. Then we predict the type for a relationship in the test set as the type with the maximal trust score.

Our methods: include eTrust and eTrust-s. They consider both attribute features and dyadic and triadic correlation features. Our methods can be viewed as semi-supervised methods, as we use both the labeled and unlabeled relationships. When we compare with the unsupervised methods, for each trust type, we predict the probability of this type for each relationship in the test set and rank all the relationships based on their probabilities.

Hyperparameters. The threshold to select the most trusted neighbors in TT is set to 0.1, and the parameter to balance the initial values and inferred values in LS is set to 0.5. We set batch size as 1000 and learning rate as 1.0 in the proposed eTrust-s.

5.1.3 *Evaluation Measures*

To quantitatively evaluate the proposed methods, we consider the following measurements.

Accuracy Performance. We use precision, recall, f1 and accuracy to evaluate the classification performance of the supervised/semi-supervised methods. For Alibaba and Advogato with multiple labels, we calculate the metrics for each label, and use their unweighted means as the combined metrics. To evaluate the ranking performance of the unsupervised methods, we use precision at top k%, where k changes from 1 to 10 with interval 1 and from 10 to 100 with interval 10, as the evaluation measure, which is well-adopted to evaluate link prediction tasks [6], [18], [39], [40]. For Alibaba and Advogato with multiple labels, we evaluate the metric for each trust type separately.

Efficiency Performance. We use the execution time of model learning to show the speedup of eTrust-s comparing with eTrust.

Application. We apply the trust relationships inferred from Ali-large into two real applications in Alibaba Group, to demonstrate the effectiveness of the learning results.

The algorithms are implemented using C++ and all experiments except those on Ali-large are performed on an Enterprise Linux Server with 24 Intel(R) Xeon(R) CPU cores (E5-2630 2.30GHz) and 100GB memory. The algorithm performed on Ali-large is distributed and performed on ODPS, a distributed platform in Alibaba Group. All codes and part of the datasets are publicly available[7].

7. https://github.com/cenyk1230/eTrust

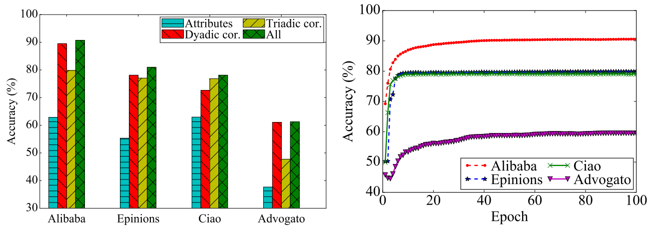
5.2 Performance Analysis

Accuracy Performance. Table 6 shows the overall prediction performance of the supervised, semi-supervised and proposed methods. In terms of accuracy, the proposed eTrust and eTrust-s achieve 1.7-32.3% improvement over all the baseline methods. Supervised methods including LR, SVM and RF only consider the attribute features of the labeled relationships but ignore the unlabeled relationships, thus the network correlations between the relationships cannot be incorporated. Semi-supervised methods LP and LS leverage the network correlations between labeled and unlabeled relationships, but perform almost the same as the supervised method RF on Ciao, and even worse than RF on Epinion and Advogato, because they ignore attribute features. Nevertheless, they perform much better than the supervised methods on Alibaba, because nearby relationships are much more likely to be the same type on Alibaba than on the other three datasets, which is more compatible with the propagation assumption of LP and LS. The proposed eTrust and eTrust-s not only consider attribute features, but also incorporate all kinds of type-based dyadic and triadic correlations between labeled and unlabeled relationships as features, and outperform all the baseline methods. Moreover, although eTrust-s infers marginal probabilities of relationships' types locally instead of globally by eTrust, eTrust-s achieves a comparable performance to eTrust on all the datasets.

Figure 3 presents the curve of precision at top k% for unsupervised methods and the proposed methods. We present the results of one trust type on Alibaba and Advogato for instance. Our methods eTrust and eTrust-s outperform all the unsupervised methods significantly. Particularly, the precision of top k% of eTrust and eTrust-s is close to 100% at the very beginning and then gradually falls to the proportion of positive instances in the whole test set with the increase of k. The results indicate that the proposed methods can rank most of the positive instances before the negative instances. However, the performance of some baseline methods such as CN, AA, JC and TT fluctuate a lot, especially on the datasets of Epinions and Ciao, which indicates a lot of the positive instances are ranked behind the negative instances at the beginning by these methods. Thus the best performance is achieved late. The improvement on Alibaba is much larger than that on the other three datasets, because compared with the networks of Epinion, Ciao or Advogato, Alibaba's network is extremely sparse (Graph density is only 5.41E-5), making it particularly difficult for the unsupervised methods to propagate trust scores along the network.

Efficiency Performance. We compare the efficiency performance of eTrust and eTrust-s, and show the CPU time required for model learning in Table 5. The proposed eTrust-s typically achieves a significant reduction of CPU time. On the four datasets, eTrust-s obtains a speedup from 1000+ to 3000+ times compared with eTrust.

Factor Analysis. We conduct an analysis on the effect of attribute features, dyadic correlation features and triadic correlation features independently. The results in Figure 4(a) show that network correlation features make more contributions on predicting trust than attribute features on all the datasets. We also find that the dyadic correlation features take much more effect than the triadic correlation features on Alibaba and Advogato, but the phenomenon is different on Epinions and Ciao. This is because the density of Alibaba and Advogato is 5.41E-5 and 3.00E-3 respectively, which is much smaller than that of Epinions and Ciao, whose density is 0.134 and 0.129 respectively. Besides, Alibaba and Advogato have more relationship types, leading to much more types of triadic structures. Thus for each relationship in Alibaba and Advogato, the available triadic correlation features of each type are much sparse, increasing the difficulty of learning the effect of triadic correlation features.



(a) Factor Analysis (b) Convergence Analysis

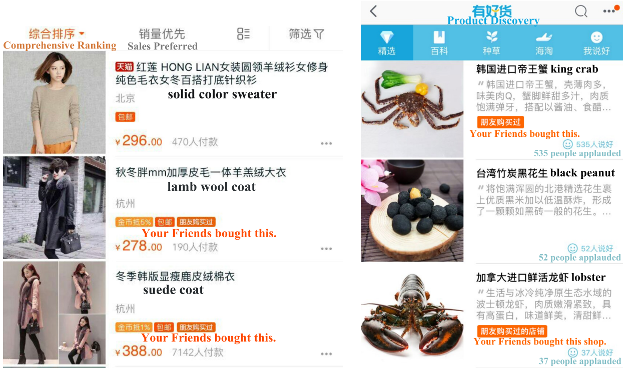
Fig. 4. Factor and convergence analysis.

Convergence Analysis. We analyze the convergence property of eTrust-s. The results in Figure 4(b) demonstrate eTrust-s can converge quickly within 50 epochs on all the four datasets. Particularly, it only takes less than 10 epochs to converge on Epinions and Ciao.

5.3 Application

We apply the trust relationships inferred from Ali-large by eTrust-s into two real products of Alibaba Group — Taobao product search and Taobao product discovery. The model eTrust takes more than several hours to perform one iteration on such a big dataset and cannot finish in an available time. The model eTrust-s only takes about 8 hours for model learning and inference, which achieves more than 2000× speed up comparing with eTrust. Figure 5 presents example results of Taobao product search and Taobao product discovery with the products purchased by trusted friends ranked higher. We conduct A/B Testing about two weeks on two methods. One is CF, a collaborative filtering method which is used to rank personalized products for a user. The other is eTrust-s, which is used to rank ahead the products purchased by the trusted neighbors inferred by the proposed eTrust-s from Ali-large. Although we inferred multi-typed trust relationships, we aggregate different trust types together and do not distinguish them. We plan to distinguish different trust types when conducting social recommendation in the future.

We evaluate the search results by three metrics including *return rate* which measures the proportion of returning



(a) Taobao product search (b) Taobao product discovery

Fig. 5. Applications of trust relationships in Taobao — Alibaba E-Commerce platform.

behaviors after purchasing a product, *poor rating rate* which measures the proportion of the purchase records that receive poor ratings from users, and *medium rating rate*. Specifically, we evaluate the metrics reduced by the applied method upon the previous results before applying the method. The three metrics are reduced by 36.78%, 54.55% and 57.38% respectively by eTrust-s, and the results significantly outperform those of CF by 30.09%, 45.45% and 42.08% respectively. It indicates that when ranking ahead the products of trusted neighbors, the search results can help users accelerate their decision process, and also improve users' purchase experience. In another application of Taobao product discovery, an addition metric — *gross merchandise volume* is improved by 2.75% when recommending the products purchased by the trusted neighbors, which implies that users highly trust the recommendations from their acquaintances.

6 Conclusion

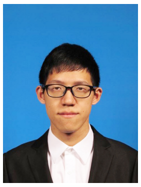
This paper studies how to infer trust relationships to facilitate Alibaba's E-Commerce business. We formalize trust into four-typed acquaintances and conduct an analysis on all kinds of type-based dyadic and triadic network corrections among relationships. Then we propose a novel method namely eTrust by incorporating the discovered correlation patterns into a factor graph model. Experimental results on four genres of real-world datasets show that the proposed method significantly outperforms comparison methods. However, the limitation of the proposed eTrust method is the potential inefficiency when dealing with large graphs. An approximate model eTrust-s is proposed to address the limitation, but may hurt the accuracy. In a number of practical applications, as we tested, eTrust-s performs very well, at least as well as eTrust. Our experimental results show that when dealing with large networks, eTrust-s can achieve $> 2000\times$ efficiency speedup, while guaranteeing a comparable accuracy to eTrust. A/B testings on Taobao product search/discovery further confirm the business value of the study. But it also shows that the performance of the proposed models is impacted when the network is too sparse, as the major improvement is caused by the network correlation features. We will further investigate how to reduce the impact caused by sparsity. Besides, people may trust others on electronics but not fashion [34], which is also important on social recommendations. We will study how to infer this multi-aspect trust in future works.

Acknowledgments. This work is supported by Development Program of China (2016QY01W0200), National Key R&D Program of China (2018YFB1004401), NSFC for Distinguished Young Scholar (61825602) and NSFC under the grant No. 61532021, 61772537, 61772536, 61702522, 61836013.

References

[1] Black friday benchmark report: 2012 the year of the mobile shopper. *https://www.ibm.com/developerworks/community/blogs/e874ec4d-2a29-41a2-8fdd-16babe9d4d21/entry/Black_Friday_Benchmark_Report_2012_the_year_of_the_mobile_shopper?lang=zh*, 2012.
[2] S. Adalı. *Modeling trust context in networks*. Springer, 2013.
[3] L. A. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, 2003.
[4] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *Proceedings of 4th ACM International Conference on Web Search and web Data Mining*, pages 635–644, 2011.
[5] D. Crandall, D. Cosley, D. Huttenlocher, J. Kleinberg, and S. Suri. Feedback effects between similarity and social influence in online communities. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 160–168, 2008.
[6] Y. Dong, J. Zhang, J. Tang, N. V. Chawla, and B. Wang. Coupledlp: Link prediction in coupled networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 199–208, 2015.
[7] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9(Aug):1871–1874, 2008.
[8] J. A. Golbeck. Computing and applying trust in web-based social networks. 2005.
[9] A. Goyal, F. Bonchi, and L. V. Lakshmanan. Learning influence probabilities in social networks. In *Proceedings of 3rd ACM International Conference on Web Search and web Data Mining*, pages 241–250, 2010.
[10] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *Proceedings of the 13th International Conference on World Wide Web*, pages 403–412, 2004.
[11] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the 4th ACM Conference on Recommender Systems*, pages 135–142, 2010.
[12] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.
[13] N. Korovaiko and A. Thomo. Trust prediction from user-item ratings. *Social Network Analysis and Mining*, 3(3):749–759, 2013.
[14] P. F. Lazarsfeld, R. K. Merton, et al. Friendship as a social process: A substantive and methodological analysis. *Freedom and control in modern society*, 18(1):18–66, 1954.
[15] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the 19th International Conference on World Wide Web*, pages 641–650, 2010.
[16] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Signed networks in social media. In *Proceedings of the 28th SIGCHI Conference on Human Factors in Computing Systems*, pages 1361–1370, 2010.
[17] D. Liben-Nowell and J. M. Kleinberg. The link-prediction problem for social networks. *Journal of American Society for Information Science and Technology*, 58(7):1019–1031, 2007.
[18] R. Lichtenwalter, J. T. Lussier, and N. V. Chawla. New perspectives and methods in link prediction. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 243–252, 2010.

[19] H. Liu, E.-P. Lim, H. W. Lauw, M.-T. Le, A. Sun, J. Srivastava, and Y. Kim. Predicting trusts among users of online communities: an epinions case study. In *Proceedings of the 9th ACM Conference on Electronic Commerce*, pages 310–319, 2008.
[20] T. Lou, J. Tang, J. E. Hopcroft, Z. Fang, and X. Ding. Learning to predict reciprocity and triadic closure in social networks. *ACM Transactions on Knowledge Discovery From Data*, 7(2), 2013.
[21] H. Ma, M. R. Lyu, and I. King. Learning to recommend with trust and distrust relationships. In *Proceedings of the 3rd ACM Conference on Recommender Systems*, pages 189–196, 2009.
[22] P. MacCarron, K. Kaski, and R. Dunbar. Calling dunbar's numbers. *arXiv preprint arXiv:1604.02400*, 2016.
[23] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, pages 467–475, 1999.
[24] V.-A. Nguyen, E.-P. Lim, J. Jiang, and A. Sun. To trust or not to trust? predicting online trusts using trust antecedent framework. In *Proceedings of the 9th IEEE International Conference on Data Mining*, pages 896–901, 2009.
[25] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 701–710, 2014.
[26] R. D. Putnam. *Bowling alone: The collapse and revival of American community*. Simon and Schuster, 2001.
[27] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*.
[28] J. Tang, Y. Chang, and H. Liu. Mining social media with social theories: a survey. *ACM SIGKDD Explorations Newsletter*, 15(2):20–29, 2014.
[29] J. Tang, H. Gao, X. Hu, and H. Liu. Exploiting homophily effect for trust prediction. In *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*, pages 53–62, 2013.
[30] J. Tang, H. Gao, H. Liu, and A. D. Sarma. etrust: Understanding trust evolution in an online world. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012.
[31] J. Tang, X. Hu, and H. Liu. Social recommendation: a review. *Social Network Analysis and Mining*, 3(4):1113–1133, 2013.
[32] J. Tang and H. Liu. Trust in social media. *Synthesis Lectures on Information Security, Privacy, & Trust*, 10(1):1–129, 2015.
[33] J. Tang, T. Lou, J. Kleinberg, and S. Wu. Transfer learning to infer social ties across heterogeneous networks. *ACM Transactions on Information Systems*, 34(2):7, 2016.
[34] J. Tang, J. Sun, C. Wang, and Z. Yang. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 807–816, 2009.
[35] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: extraction and mining of academic social networks. In *SIGKDD'08*, pages 990–998, 2008.
[36] W. Tang, H. Zhuang, and J. Tang. Learning to infer social ties in large networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 381–397, 2011.
[37] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks. In *Proceedings of the 6th International Conference on Learning Representations*, 2018.
[38] C. Wang, V. Satuluri, and S. Parthasarathy. Local probabilistic models for link prediction. In *Proceedings of the 7th IEEE International Conference on Data Mining*, pages 322–331, 2007.
[39] D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A.-L. Barabasi. Human mobility, social ties, and link prediction. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and pages=1100–1108, year=2011,*.
[40] Y. Yang, R. N. Lichtenwalter, and N. V. Chawla. Evaluating link prediction methods. *Knowledge and Information Systems*, 45(3):751–782, 2015.
[41] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in neural information processing systems*, pages 321–328, 2004.
[42] J. Zhu, H. Ma, C. Chen, and J. Bu. Social recommendation using low-rank semidefinite program. In *Proceedings of the 25th Conference on Artificial Intelligence*, pages 158–163, 2011.
[43] X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning*, pages 912–919, 2003.



Yukuo Cen is a master candidate in the Department of Computer Science and Technology, Tsinghua University. He got his bachelor degree in Computer Science and Technology from Tsinghua University. His research interests include social influence and graph embedding.

Jing Zhang received the master and PhD degree from the Department of Computer Science and Technology, Tsinghua University. She is an assistant professor in Information School, Renmin University of China. Her research interests include social network mining and deep learning.

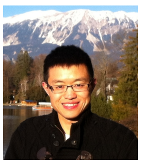
Gaofei Wang is working as the Staff Engineer in Alibaba Group. His interests span social tie prediction, information retrieval and vector retrieval. He received his Master degree from the Department of Computer Science and Technology, Harbin Engineering University in 2006. From then on, he has been working in Alibaba Group.

Yujie Qian is a PhD student at MIT Computer Science and Artificial Intelligence Laboratory, working with Prof. Regina Barzilay. He received his B.Eng. from the Department of Computer Science and Technology at Tsinghua University in 2017. His research interests include natural language processing and machine learning.

Chuizheng Meng is a PhD student in the University of Southern California. He is interested in applying machine learning and data mining algorithms to deal with special structure including relational data and time series.

Zonghong Dai is working as the Senior Staff Engineer in Alibaba Group. His interests span social tie prediction, information retrieval, vector retrieval and AI technology. He is the core originator of Pailitao Image search engine.

Hongxia Yang is working as the Senior Staff Data Scientist and Director in Alibaba Group. She got her PhD degree in Statistics from Duke University in 2010. She has published over 30 over papers and held 9 filed/to be filed US patents and is serving as the associate editor for Applied Stochastic Models in Business and Industry.



Jie Tang received the PhD degree from Tsinghua University. He is full professor in the Department of Computer Science and Technology, Tsinghua University. His main research interests include data mining, social network, and machine learning. He has published over 200 research papers in top international journals and conferences.