Noname manuscript No.
(will be inserted by the editor)

Trust Prediction from User-Item Ratings

Nikolay Korovaiko · Alex Thomo

Received: date / Accepted: date

Abstract Trust relationships between users in various online communities are notoriously hard to model for computer scientists. It can be easily verified that trying to infer trust based on the social network alone is often inefficient. Therefore, the avenue we explore is applying Data Mining algorithms to unearth latent relationships and patterns from background data. In this paper, we focus on a case where the background data is user ratings for online product reviews. We consider as a testing ground a large dataset provided by Epinions.com that contains a trust network as well as user ratings for reviews on products from a wide range of categories. In order to predict trust we define and compute a critical set of features, which we show to be highly effective in providing the basis for trust predictions. Then, we show that state-of-the-art classifiers can do an impressive job in predicting trust based on our extracted features. For this, we employ a variety of measures to evaluate the classification based on these features. We show that by carefully collecting and synthesizing readily available background information, such as ratings for online reviews, one can accurately predict social links based on trust.

Keywords Trust Prediction · Trust Inference · Epinions · Data Mining

Nikolay Korovaiko
University of Victoria
3800 Finnerty Road,Victoria BC V8P 5C2,Canada
E-mail: nikolayk@uvic.ca

Alex Thomo
University of Victoria
3800 Finnerty Road,Victoria BC V8P 5C2,Canada
E-mail: thomo@cs.uvic.ca

1 Introduction

With the explosive growth in popularity of social networks and e-commerce systems, users are constantly in interaction with each other. The trust factor plays an important role in initiating these interactions and building higher-quality relationships between the users.

Consider some examples. We are more willing to buy an item from a particular seller on E-Bay or Amazon, if either we or our friends had positive experience with that seller in past. On the other hand, we are reluctant to engage in any relationship with strangers. On freelance websites, trust means fruitful agreements between a professional and an employer. Dating services might try to leverage users' preferences to help their users find a perfect match.

Our attitudes towards trust are often very different and individual. One might believe that a particular seller on E-Bay provides an excellent service, even though this seller sometimes delays shipping by a week. For another person any delay might be unacceptable. Trust-aware systems can help users make the right choices and have relationships that lead to positive outcomes. Even though trust takes many different meanings and highly depends on the context in which users interact with each other, it can be shown that trust can be approximated from other relationships. In online communities, users interact with each other in many ways. In Epinions for example (which is the focus of this paper), the active users participate in the discussions that grow around various products and write reviews on these products. The rest of the user community comments and rates the reviews. Additionally, users can specify whom they trust. These trust connections constitute the user's trust network. The problem we study here is how to predict these trust links. This is an impor-

tant problem which, when solved effectively, enhances the user online experience by connecting him/her to peers who share the same interests and values. The users can rely on the input from their peers or trustees to form their own opinion about a particular product much faster and easier.

There has been extensive research leveraging trust links to produce more accurate rating predictions (cf. Massa and Avesani [2009], Jamali and Ester [2009], Chowdhury et al [2009]). However, for the inverse problem we study here, predicting trust links from product ratings, there is considerably less research done so far. The inverse problem is equally important. Without reliable trust prediction and recommendation, the users' trust networks will not grow fast, especially in an environment like Epinions, where the users do not in general know each other personally as friends in real life. Limited trust networks do not provide for an enhanced online experience, and may alienate the users from the system.

The traditional way of predicting trust links is based on the notion of "trust propagation" that uses only the existing trust network to predict new trust links. However, the ratings that users have given to products or reviews provide us with rich background information. Tapping into this wealth of information should positively affect the quality of predictions. Furthermore, we might be able to infer trust relationships in cases where the traditional trust propagation algorithms fail. For instance, by using user ratings for online reviews, we might be able to find users who have similar preferences and thus would probably trust each other even though, in terms of the current trust graph, they appear to be quite far from each other. In this work, we treat trust prediction as a classification problem and focus on the following aspects of trust prediction.

- We explore trust patterns prevailing in the Epinions online community. We show that these patterns can be approximated by a set of quantitative features. Classifiers trained on our features show a 5-20% improvement in performance (e.g. precision, recall, F-measure, ROC Area, etc) over similar approaches Liu et al [2008] and Nguyen et al [2009].
- When it comes to the trust prediction problem, user similarity tends to be neglected. Our experiments show that user similarity features correlate strongly with the users' trust decisions: two of our top 10 most important features are based on user similarity. We explore more complex user similarity features than the ones suggested in previous research. In particular, we experiment with using a Jaccard Similarity Index for partitioned sets of higher and lower ratings. The partitions allow one to detect conflicts in users' preferences, which turns out to be a very important feature for trust prediction. We also address the sparseness problem of the ratings by computing user similarities with respect to categories and reviewers they have considered and rated. Computing such user similarities also captures the implicit classification of the ratings, as the reviewers may represent trends spanning several categories. We also suggest a user similarity feature for reviewers specifically. The feature reveals whether the reviewers share some common interests.
- Rater-Reviewer interactions are quite useful for trust prediction. We attempt to make rater-reviewer features more robust to individual user biases by applying various Data Mining techniques. We also suggest several features based on those ratings that have been anonymized. To the best of our knowledge, anonymized ratings tend to be largely under-used for trust prediction. Additionally, we explore alternative ways to incorporate leniency and reputation.
- Lastly, we suggest a personalized trust prediction model that infers trust between users based on the opinions of the closest trustees of the truster towards the trustee. The model also discriminates between various user interactions.

The rest of this paper is organized as follows. Section 2 provides a comprehensive overview of the research that has been conducted on trust prediction. Section 3 discusses the features proposed for trust prediction and also introduces our personalized trust prediction model. In Section 4 we rank our features and also compare our approach against similar ones. Section 5 summarizes our findings and outlines future research on trust prediction.

2 Related Work

Jennifer Golbeck was one of the first pioneers to research the problem of trust prediction from a Computer Science perspective. In Golbeck [2005], she discusses various properties of trust, such as transitivity, composability, and asymmetry, and proposes algorithms for inferring binary and continuous trust values from trust networks, based on trust propagation. Kuter and Golbeck [2007] suggest another trust inference algorithm called *Sunny*. The algorithm uses a probabilistic sampling technique to estimate the confidence in the trust information from some designated sources.

Guha et al [2004] propose iterative methods for inferring trust ratings. They define four one-step trust or distrust propagations in terms of basic matrix operations. Algorithms are suggested to combine the atomic

steps in order to generate a final belief matrix that is used for trust prediction. The algorithms employ different strategies to deal with rounding and the large numbers of iterations.

An efficient trust propagation algorithm is suggested by Massa and Avesani [2005]. To compute the trust rating for a particular sink, its neighbours are first filtered to exclude untrustworthy members whose trust ratings are less than a threshold. Then, a weighted average is computed and assigned to the sink. The algorithm starts from a source and recursively computes the trust ratings for the nodes it discovers until the rating for the sink is computed.

Sherchan et al [2011] suggest a temporal Hidden Markov Model for reputation prediction. The model has five states and each state is represented by four hidden factors. To incorporate temporal sensitivity into a basic model, the authors suggest to remove older data and add more recent one in each iteration.

Ma et al [2009] derive various features from writer-reviewer interactions and use the features in personalized and cluster-based classification methods. They train one classifier for each user using user-specific training data. Their cluster-based method constructs user clusters, which are then used to train a personalized classifier for a particular user.

Skopik et al [2009] focus on the issue of bootstrapping. They introduce trust mirroring and trust teleportation in order to reliably infer trust relationships from datasets with a very few or no trust links. First, the authors build hierarchical tag clusters to group similar and synonymous tags. Trust mirroring allows one to predict a trust link based on whether users use tags in the same way. On the other hand, trust teleportation uses an existing trust link between two users and applies trust mirroring similarity in order to make a trust prediction.

Yet another approach to trust prediction, which we also follow in this paper, is to treat it as a classification problem, allowing one to leverage the rich repertoire of existing Data Mining algorithms. In this scenario, raw data is often pre-processed. A motivation behind preprocessing is to transfer some higher-level knowledge that one has about the data to a classification algorithm directly. The approach has been shown to be quite effective by the following research. Liu et al [2008] develop a taxonomy of user relationships for the Epinions dataset. This taxonomy is used to obtain an extensive set of simple features which is in turn employed for training Naive Bayes and Support Vector Machines (SVM) classifiers. However, one should note that it is not always feasible to employ the overwhelmingly large number of features suggested by Liu et al [2008]. Moreover, some of the features can be quite naturally combined into a single one resulting in more accurate predictions. Nguyen et al [2009] derive several trust prediction models from a well-studied Trust Antecedent Framework used in Management Science. The framework captures the following three factors: ability, benevolence and integrity. The authors approximate each factor through a set of quantitative features, which are then used for training an SVM classifier. Borzymek et al [2009] suggest a set of five user similarity features. The first three features are graph-based and capture the incoming and outgoing edges for a pair of users. The last two features capture the number of reviews of a prospective trustee, and the number of the rated items the users have in common.

Lastly, Noor and Sheng [2011] and Sinclaire et al [2010] focus on the *trustworthiness* of prediction and the impact it has on consumers. Noor and Sheng [2011] suggest to compute the trustworthiness as a sum of feedbacks weighed by their trust credibilities. The trust credibilities are in turn computed from two major components: Feedback Density and Majority Consensus. Majority Consensus measures how well the feedbacks by a particular set of consumers are aligned with the feedback majority. Feedback Density penalizes services that receive their feedbacks from a smaller number of unique consumers.

3 Trust Prediction Model

We strongly believe that understanding the direct and (often) indirect interactions between users is crucial for producing high quality trust ratings. This understanding enables one to accurately approximate the interactions with a set of quantitative features. This work is a study into the interactions that prevail in online communities. We also propose a way to rank features that allows one to tell which features are more important for trust prediction. We suggest multiple features that can be roughly split into two large categories: user similarity and rater-reviewer interactions. One can use our features to complement the ones developed in previous research, but we will show that our features favorably compare against the ones suggested by Liu et al [2008] and Nguyen et al [2009]. We also suggest a Personalized Trust Prediction model that leverages the topology of a trust graph.

Before we dive into the discussion of the features, let us review a few key factors affecting the decision of a particular user to trust another one as our features are directly based on these. First, both users might simply have very similar preferences. In other words, they tend to like the same items. Second, the user can trust another if he decides that the person is a good reviewer

who writes high quality reviews on different products. Third, the user might think of the other person as being a good review critic. Finally, both users might be friends. In the latter case, there is typically a mutual trust link between the users, even if they do not have that many things in common.

We can categorize each factor as either a rater-reviewer interaction or a similarity factor. The former group of factors is quite important providing for accurate recommendations. However, user similarity factors also appear to play an important role in establishing trust relationships between the members of the Epinions community. About 41% (292,793) of trust links come from pairs of users who do not have a rater-reviewer interaction. Over 90% of trust links come from pairs of users who have rated at least one item similarly.

We consider the following user similarity interactions and derive features based on each factor:

1. u and v give similar ratings to the reviews they read
2. u and v are interested in similar categories of products
3. u and v produce reviews in the same categories that interest them
4. u and v rate the reviews produced by the same reviewers
5. u and v have the same trustees.

We also include features based on rater-reviewer interactions:

1. u gives high ratings to reviews produced by v
2. u anonymizes a considerable number of ratings for reviews produced by v
3. v is a reputable reviewer

A few rater-reviewer features that we suggest leverage the data on hidden ratings.

In general, we achieve an improvement of 5-20% in various performance metrics (e.g. precision, recall, F-measure and ROC Area) over other competing approaches.

Finally, let us introduce the notation that is used throughout this discussion. We use the terms: item and review interchangeably. Let u, v, y be users. We denote by

U	the set of users
I_u	the set of items rated by u
$I_{u,r}$	the set of items rated r (where $r \in \{1,2,3,4,5\}$) by u
$I_{u,c}$	the set of items in category c rated by u
$I_{u,y}$	the set of reviews (items) produced by y and rated by u
J_v	the set of reviews (items) produced by v
C_u	the set or multiset (depending on the feature) of categories of the items in I_u
D_u	the set or multiset (depending on the feature) of categories of the reviews (items) produced by u
Y_u	the set or multiset (depending on the feature) of reviewers (users) who have produced the reviews (items) in I_u
T_u	the set of trustees of user u, *i.e.* those users that u trusts.

For simplicity we will denote by $r_{u,i}$ a rating (u, i, r) that a user u gives for item i. This also reflects the fact that for a given user and a given item there can be not more than one rating. Let us also remind the reader that we treat trust prediction as a classification problem, that is, for each ordered pair (u, v) of users, a new value called class (trust or distrust) has to be assigned. There is a rich repertoire of classifier algorithms available for this classification problem. In our experiments, we choose to use Random Forests (RF) and Support Vector Machines (SVM).

Parameter 1: *u and v give similar ratings to the reviews they read.*

If two users have similar preferences (i.e. u likes the same items as v), one of the users will trust the other one's recommendations. In other words, there will be at least one trust link between this pair of the users. The more similar the users' preferences the more probable trust link is. There are multiple metrics for measuring user similarity given rating information. Our first feature uses Pearson Correlation. PC is widely adopted by the scientific community, as it allows one to compare the ranking system of one user to the ranking system of another. In other words, it reconciles more critical users and less critical users. PC also provides a single score in the $[-1, 1]$ interval. Given that $r_{u,i}$ is a rating given by u to the item i we also introduce $\bar{r}_u$ and $\bar{r}_v$ to denote the average ratings by u and v, correspondingly. Then, $f_{1,a}$ is defined as follows:

$$f_{1,a} = \frac{\sum_{i \in I_u \cap I_v} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_u \cap I_v} (r_{vi} - \bar{r}_v)^2}}.$$

Experimenting with various metrics we noticed that it could be very beneficial to partition ratings into higher and lower segments. The higher segment includes the ratings of four and five, whereas the lower one includes all the ratings of one and two. The ratings of four or five are a strong indicator of user likes. Alternatively, ratings of one or two are a strong indicator of user dislikes. Intuitively, when u and v have a relatively significant number of compatible partisan ratings, their likes and dislikes are aligned. On the other hand, when u and v have incompatible partisan ratings, e.g. u gives a rating of one whereas v gives a rating of five, their preferences exhibit a conflict. We employ the Jaccard Similarity Index to measure the extent to which users agree (b and c) and disagree (d and e) in their preferences. JCI is much easier to compute, yet the features based on JCI yield fairly accurate results. Let

$I_{u,\uparrow} = I_{u,5} \cup I_{u,4}$, $I_{u,\downarrow} = I_{u,1} \cup I_{u,2}$.

$I_{v,\uparrow}$ and $I_{v,\downarrow}$ are defined, analogously. Now we define

$$f_{1,b} = \frac{|I_{u,\uparrow} \cap I_{v,\uparrow}|}{|I_{u,\uparrow} \cup I_{v,\uparrow}|}, \;\; f_{1,c} = \frac{|I_{u,\downarrow} \cap I_{v,\downarrow}|}{|I_{u,\downarrow} \cup I_{v,\downarrow}|}$$
$$f_{1,d} = \frac{|I_{u,\uparrow} \cap I_{v,\downarrow}|}{|I_{u,\uparrow} \cup I_{v,\downarrow}|}, \;\; f_{1,e} = \frac{|I_{u,\downarrow} \cap I_{v,\uparrow}|}{|I_{u,\downarrow} \cup I_{v,\uparrow}|}.$$

Parameter 2: *u and v are interested in similar categories of products.*

Typically, the user similarity features based on rating information give the best results, as the ratings are user interactions at the smallest granularity level. On the other hand, the rating information is very sparse. Consider, a real-world example. There might be the pair of users who both enjoy reading sci-fi reviews. However, the users rarely rate the same reviews, as there is a huge number reviews written on scientific literature. This example motivates us to compute user similarity based on categories, which is a higher granularity level. The set of features computed on category information alleviates the sparseness problem. The simplest approach to estimate the user similarity based on categories is to compute JCI over the categories in which both users rated reviews. We start with

$$f_{2,a} = \frac{|C_u \cap C_v|}{|C_u \cup C_v|}$$

However, this feature might be too coarse for certain scenarios. The features below also take into consideration the number of the ratings and average rating, respectively. We compute the number of ratings ($f_{2,b}$) and average rating ($f_{2,c}$) in each category for both users. Then we use PC to estimate the user similarity over the numbers received from the first step. Formally, we have

$$f_{2,b} = \sum_{c \in C_u \cap C_v} \left(|I_{u,c}| - \frac{|I_u|}{|C_u|}\right)\left(|I_{v,c}| - \frac{|I_v|}{|C_v|}\right) \cdot \frac{1}{\sqrt{\sum_{c \in C_u \cap C_v} \left(|I_{u,c}| - \frac{|I_u|}{|C_u|}\right)^2}} \cdot \frac{1}{\sqrt{\sum_{c \in C_u \cap C_v} \left(|I_{v,c}| - \frac{|I_v|}{|C_v|}\right)^2}}$$

(In this definition, C_u and C_v are considered as sets.) and

$$f_{2,c} = \sum_{c \in C_u \cap C_v} (r_{u,c} - \hat{r}_u)\,(|r_{v,c}| - \hat{r}_v) \cdot \frac{1}{\sqrt{\sum_{c \in C_u \cap C_v} (r_{u,c} - \hat{r}_u)^2}} \cdot \frac{1}{\sqrt{\sum_{c \in C_u \cap C_v} (r_{v,c} - \hat{r}_v)^2}}$$

where

$$r_{u,c} = \frac{\sum_{i \in I_{u,c}} r_{u,i}}{|I_{u,c}|}, \;\; \hat{r}_u = \frac{\sum_{c \in C_u} r_{u,c}}{|C_u|}$$
$$r_{v,c} = \frac{\sum_{i \in I_{v,c}} r_{v,i}}{|I_{v,c}|}, \;\; \hat{r}_v = \frac{\sum_{c \in C_v} r_{v,c}}{|C_v|}.$$

(For this definition as well, C_u and C_v are considered as sets.)

Parameter 3: *u and v produce reviews in the same categories that interest them.*

We can also compute a so-called reviewer similarity. Our experiments showed that even simple JCI over the categories in which both reviewers have reviews yield good results. Formally,

$$f_3 = \frac{|D_u \cap D_v|}{|D_u \cup D_v|}.$$

In this definition, D_u and D_v are considered as multisets and epresent the sets of categories in which u and v produce reviews

Parameter 4: *u and v rate reviews produced by the same reviewers.*

Another indication of similar user preferences is when both users favor the reviews written by the same reviewers. Typically, if a user likes a reviewer, the user gives higher ratings to reviews from that reviewer. Again this parameter alleviates the sparseness of the ratings. However, it is quite different from parameter 2, as it uses an aggregation which is based on reviewers. We assume that the average rating given by a user to the reviews from a reviewer ($r_{u,y}$ and $r_{v,y}$) reflects the user's

preferences towards the reviewer. We employ the Pearson correlation to compute the similarity between the two sets of average ratings given by u and v to reviewers. Formally,

$$f_4 = \frac{\sum_{y \in Y_u \cap Y_v} (r_{u,y} - \check{r}_u)(|r_{v,y}| - \check{r}_v)}{\sqrt{\sum_{y \in Y_u \cap Y_v} (r_{u,y} - \check{r}_u)^2} \sqrt{\sum_{y \in Y_u \cap Y_v} (r_{v,y} - \check{r}_v)^2}}$$

where

$$r_{u,y} = \frac{\sum_{i \in I_{u,y}} r_{u,i}}{|I_{u,y}|}, \quad \check{r}_u = \frac{\sum_{y \in Y_u} r_{u,y}}{|Y_u|}$$

$$r_{v,y} = \frac{\sum_{i \in I_{v,y}} r_{v,i}}{|I_{v,y}|}, \quad \check{r}_v = \frac{\sum_{y \in Y_v} r_{v,y}}{|Y_v|}.$$

In this definition, Y_u and Y_v are considered as sets.

Parameter 5: *u gives high ratings to reviews produced by v.*

If u gives high ratings to the reviews produced by v, there is typically a trust link connecting u to v. Intuitively, if we appreciate the reviewers written by a particular reviewer we tend to trust the reviewer's recommendations as well.

The basic approach that has been successfully used by Liu et al [2008] is to compute an average rating that the user gives to the reviews (items) produced by v. However, one can significantly improve on this approach by applying various techniques from rating prediction. Our first feature borrows from the *baseline predictors* technique suggested in Koren [2010]. Rather than just using the average, we compute the sum of four following components. The first component is the global average of all ratings in our sampled dataset, which we denote by $\bar{r}$. The second, third, and fourth components are the differences from the global average of the following averages:

- the average of all ratings given to the items produced by v, denoted by $\check{r}_v$
- the average of all ratings u gives, denoted by $\bar{r}_u$
- the average of all ratings that u gave to the items produced by v, denoted–similarly as for Parameter 4–by $r_{u,v}$.

The baseline predictors offer at least two advantages over the standard average. First, the technique removes the users' biases making it possible to compare the ratings of two users directly. Second, it makes the feature resilient to outliers and the sparseness problem. For example, if u only rated a couple of the v's reviews we could still make a reliable trust prediction based on the popularity of v and average trusting trends in the dataset. Formally, we have:

$$f_{5,a} = \bar{r} + (\check{r}_v - \bar{r}) + (\bar{r}_u - \bar{r}) + (r_{u,v} - \bar{r}).$$

We also experimented including these two very basic features computing the fractions of higher and lower ratings that u gives to reviews (items) produced by v.

$$f_{5,b} = \frac{|I_{u,\uparrow} \cap J_v|}{|I_{u,\uparrow}|}, \text{and } f_{5,c} = \frac{|I_{u,\downarrow} \cap J_v|}{|I_{u,\downarrow}|}.$$

Parameter 6: *u anonymizes a considerable number of ratings for reviews produced by v.*

The users mostly anonymize the lower ratings. One can explain this phenomenon by our innate reluctance to deliver bad news. If u anonymizes a considerable number of ratings for reviews produced by v, those ratings are most likely the lower ones and u probably does not trust v. However, the above is not necessarily true. Some users are simply very private. Such users hide the larger part their ratings. The considerations above naturally motivate us to split ratings into the lower and higher segments before converting the ratings into features. We start by computing the ratio of anonymized ratings u gives to the v's items,

$$\frac{|I^-_{u,v}|}{|I_{u,v}|}$$

where $I^-_{u,v} \subseteq I_{u,v}$ is the set of v's items rated anonymously by u.

We then consider the high or low ratings only, and have $\frac{|I^-_{u,v,\uparrow}|}{|I_{u,v,\uparrow}|}$ and $\frac{|I^-_{u,v,\downarrow}|}{|I_{u,v,\downarrow}|}$, where $I^-_{u,v,\uparrow}$, $I_{u,v,\uparrow}$, $I^-_{u,v,\downarrow}$, and $I_{u,v,\downarrow}$ are defined as their non-arrow counterparts, but considering the high or low ratings only.

The baseline predictors technique can be also applied to these ratios. For this, let

R the set of all ratings
R^- the set of all anonymous ratings ($R^- \subseteq R$)
$R_{\to v}$ the set of all ratings for v's items
$R^-_{\to v}$ the set of all anonymous ratings for v's items
$R_{u\to}$ the set of all u's ratings
$R^-_{u\to}$ the set of all anonymous u's ratings.

Also let $R_\uparrow$, $R^-_\uparrow$, $R_{\to v,\uparrow}$, $R^-_{\to v,\uparrow}$, $R_{u\to,\uparrow}$, $R^-_{u\to,\uparrow}$ be defined similarly as above, but with only the high ratings considered. Likewise for $R_\uparrow$, $R^-_\downarrow$, $R_{\to v,\downarrow}$, $R^-_{\to v,\downarrow}$, $R_{u\to,\downarrow}$, $R^-_{u\to,\downarrow}$, but with only the low ratings considered.

We now define

$$f_{6,a} = \frac{|R^-|}{|R|} + \frac{|R^-_{\to v}|}{|R_{\to v}|} + \frac{|R^-_{u\to}|}{|R_{u\to}|} + \frac{|I^-_{u,v}|}{|I_{u,v}|}$$

$$f_{6,b} = \frac{|R^-_\uparrow|}{|R_\uparrow|} + \frac{|R^-_{\to v,\uparrow}|}{|R_{\to v,\uparrow}|} + \frac{|R^-_{u\to,\uparrow}|}{|R_{u\to,\uparrow}|} + \frac{|I^-_{u,v,\uparrow}|}{|I_{u,v,\uparrow}|}$$

$$f_{6,c} = \frac{|R^-_\downarrow|}{|R_\downarrow|} + \frac{|R^-_{\to v,\downarrow}|}{|R_{\to v,\downarrow}|} + \frac{|R^-_{u\to,\downarrow}|}{|R_{u\to,\downarrow}|} + \frac{|I^-_{u,v,\downarrow}|}{|I_{u,v,\downarrow}|}.$$

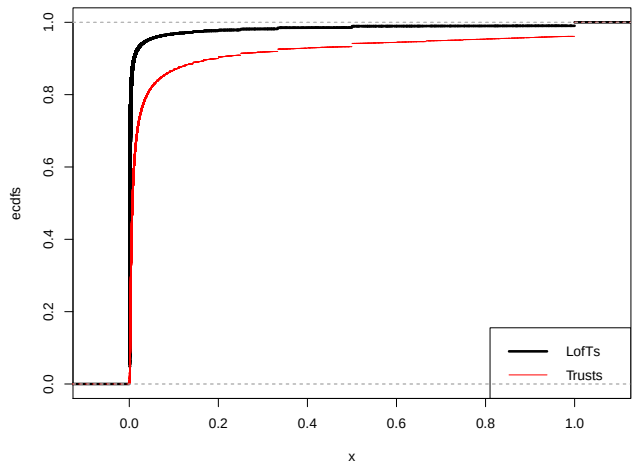

Fig. 1 The empirical cumulative distribution functions for trust and lack of trust statements for $f_{5,a}$

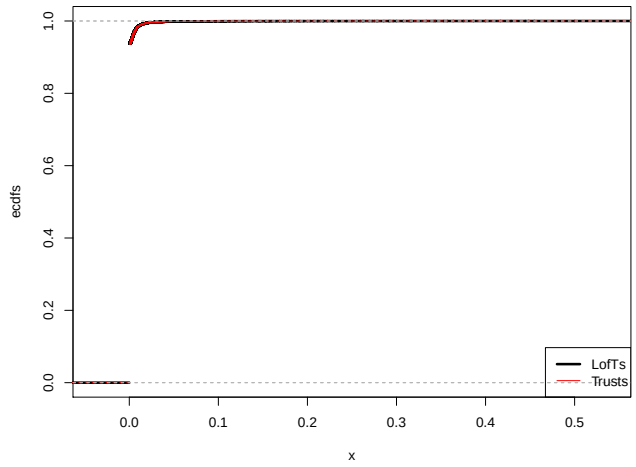

Fig. 2 The empirical cumulative distribution functions for trust and lack of trust statements for $f_{1,b}$

In $f_{6,a}$ the first component is how often, on average, users decide to keep their ratings anonymous, the second is how often, on average, v receives anonymous ratings, and the third is how often, on average, u gives anonymous ratings. Similar comments can also be made for the $\uparrow$ and $\downarrow$ components, but considering the high or low ratings only.

Parameter 7: *v is a reputable reviewer and u is a lenient rater.*

Typically, the users express their appreciation to the reviewer by giving higher ratings to his items. The more positive ratings a reviewer receives, the higher his reputation. This observation can be converted into a feature by computing the difference of the average rating given to the items produced by v from the overall average rating in the dataset.

$f_{7,a} = \check{r}_v - \bar{r}$.

On the other hand, the leniency of u also affects the trust decision. u might be very lenient comparing to an overall user leniency giving higher ratings to the reviews and trusting the reviewers more often. The leniency is computed as the difference between the average rating that u gives to the reviews and the overall average rating in the dataset.

$f_{7,b} = \check{r}_u - \bar{r}$.

The last feature we include for this parameter is equal to the difference between the average rating given to v and the average rating given by u

$f_{7,c} = \check{r}_v - \check{r}_u$.

Parameter 8: *u and v have the same trustees.*

It is important to remember that when computing the user similarity between a pair of users, we are not necessarily constrained to the user rating information only. We could also use the following intuition. If two users have the same friends, they might have similar preferences and trust each other's recommendations. The feature below computers JCI over the numbers of trustees that both users have in common.

$$f_8 = \frac{|T_u \cap T_v|}{|T_u \cup T_v|}.$$

3.1 Personalized Trust Prediction Model

We suggest a simple Personalized Trust Prediction Model that assigns a trust score to each pair (u, v). The model infers trust for some user pair (u, v) based on the trust relationships that the trustees of u indicate towards v. This approach has been widely used before. However, our model augments the approach by weighing the opinion of each trustee y in the u's trust network (i.e. y where $y \in T_u$) in order to reflect both Rater-Reviewer and User-Similarity features between u and y. If u appreciates the reviews written by y, the u's decision on initiating a trust relationship with v might be *influenced* by the y's opinion about v. The more u appreciates y the more the y's opinion influences the u's decision. Various Rater-Reviewer and User-Similarity features can be used. We consider the following weighing options:

– The average rating that u gave to the items written by y. Let us remind the reader that this feature represents the Rater-Reviewer class. This option is denoted by $score_1$.

– The extent to which users disagree in their preferences measured by $f_{1,e}$. This is the second most important feature that belongs to the User Similarity class. Similarly, the option is denoted by $score_2$.
– A linear sum of the average rating and $f_{1,e}$ denoted by $score_3$.

One can include in more features and/or try various linear combinations of those features to receive more accurate predictions. We denote the trust relationship between y and v by t_{yv}. t_{yv} is equal to 1 if $v \in T_y$ or -1, otherwise. $f_{1,e,y,v}$ is the feature $f_{1,e}$ computed for the pair (y, v).

$$score_1 = \sum_{y \in T_u} t_{yv} \cdot \bar{r}_{yv},$$

$$score_2 = \sum_{y \in T_u} t_{yv} \cdot f_{1,e,y,v},$$

$$score_3 = \sum_{y \in T_u} t_{yv} \cdot (f_{1,e,y,v} + \bar{r}_{yv}).$$

Lastly, if the score is positive, there is a trust link from u to v. The negative score indicates the absence of the link. The results for the PTP model are given in Table 2. The reader might notice that utilizing only the opinions of the trustees reduces the performance significantly. $score_1$ gives the highest precision of 0.5707, whereas $score_3$ yields the best recall of 0.3460. The scores for $score_1$ and $score_2$ again confirm that the Rater-Reviewer interactions are more important for trust prediction than the User-Similarity ones. The ROC Area for all three models is over 50%. Currently, our other approaches outperform the PTP model. However, the PTP model can be further extended or combined with other approaches as it is conceptually and computationally simple and highly customizable. Including more opinions (i.e. adding the opinions of second degree trustees) and more features might allow one to significantly improve the performance. This is a direction for future work.

4 Evaluation

4.1 Ranking Features

The research conducted by Liu et al [2008] underscores the need for a methodology enabling one to identify the features that are the most important for trust prediction. Evaluating all possible combinations of features or employing every single one is frequently considered to be infeasible. In this study we propose to employ the Kolmogorov-Smirnov test for ranking features in a decreasing order of their respective discriminatory powers.

Feature	D
$f_{5,a}$	0.5367
$f_{1,e}$	0.4615
$f_{7,b}$	0.4517
$f_{5,b}$	0.2447
$f_{6,e}$	0.1861
$f_{6,d}$	0.1623
$f_{6,b}$	0.1467
$f_{1,a}$	0.1461
$f_{6,c}$	0.0471
$f_{5,c}$	0.0463
$f_{1,d}$	0.0422
$f_{6,a}$	0.0379
$f_{7,c}$	0.0225
$f_{7,a}$	0.0113
f_3	0.0014
$f_{1,c}$	0.0013
$f_{2,a}$	0.0013
$f_{2,b}$	0.0013
f_4	0.0013
f_8	0.0013
$f_{2,c}$	0.0011
$f_{1,b}$	0.0004

Table 1 The features in the descending order of the D-statistics.

Each feature provides us with a set of values that can be naturally partitioned into trust and lack of trust sub-populations. The Kolmogorov-Smirnov test measures the extent to which the sub-populations are different. In short, the test uses the maximum vertical deviation between the two curves of the empirical distribution functions derived from the datasets. The statistics is conventionally denoted by D. Figures 1 and 2 contrast the ecdfs for trust and lack-of-trust sub-populations derived from $f_{5,a}$ and $f_{1,b}$, respectively. The reader might notice that for $f_{1,b}$ the lines are virtually indiscernible, which corresponds to a very small value of D.

Table 1 shows that the Rater-Reviewer features (e.g. $f_{5,a}$, $f_{7,b}$, $f_{5,b}$, $f_{6,e}$, $f_{6,d}$, $f_{6,b}$, $f_{6,c}$, $f_{5,c}$, $f_{6,a}$, $f_{7,c}$, $f_{7,a}$) have a much stronger discriminatory power than the User Similarity ones (e.g. $f_{1,e}$, $f_{1,a}$, $f_{1,d}$, f_3, $f_{1,c}$, $f_{2,a}$, $f_{2,b}$, f_4, f_8, $f_{2,c}$, $f_{1,b}$). Eight out of ten (e.g. $f_{5,a}$, $f_{7,b}$, $f_{5,b}$, $f_{6,e}$, $f_{6,d}$, $f_{6,b}$, $f_{6,c}$, $f_{5,c}$) *top* features from Table 1 belong to the Rater-Reviewer class, whereas only two come from the User Similarity group (e.g. $f_{1,e}$, $f_{1,a}$). Feature $f_{5,a}$, corresponding to the user leniency towards the reviewer, has the greatest discriminatory power. Surprisingly, the user leniency affects the classifier's accuracy to a much greater extent than the reviewer's reputation. The users in the Epinions community seem to not be influenced by peer pressure!

This also shows that using the complex features and applying Data Mining techniques to these features might yield significant gains. The second top feature

indicates the conflicts in the preferences between the rater and reviewer. Another interesting observation is that the features computed from the lower partisan ratings have greater discriminatory power than the features based on higher partisan ratings even though their fraction is significantly smaller than the higher ones. Users seem to exercise extreme caution and give more thought to their decision of giving a lower rating.

4.2 Experimental Design

We classify the research done on trust prediction into three categories or groups. The research in the first group (Sinclaire et al [2010], Noor and Sheng [2011]) focuses on trust credibility prediction, which is a more specialized problem than trust prediction. The second group includes standalone trust prediction algorithms (Guha et al [2004], Golbeck [2005], Massa and Avesani [2005], Ma et al [2009], Skopik et al [2009]). We decided to not evaluate our work against the first and second groups, as the algorithms are used in a different setting (e.g. the algorithms are either used to solve a more specialized problem or operate on different inputs). Developing a framework that provides one with the means for comparing such dissimilar approaches would be better suited for a review paper and is beyond of the scope of this work.

On the other hand, research works in the third group (Nguyen et al [2009], Liu et al [2008], Borzymek et al [2009]) use the following framework for trust inferring. First, the data is pre-processed and converted into a set of features that represent user interactions. Second, Data Mining classifiers are trained on the features generated in the first step and used for trust prediction. Our approach nicely fits into this family of trust prediction algorithms. We compare our work against both, Liu et al [2008] and Nguyen et al [2009]. Borzymek et al [2009] is implicitly included in Liu et al [2008].

The first model denoted by *ant8* consists of eight features derived from the Antecedent Framework Nguyen et al [2009]. The second one, *top7*, includes top seven features from Liu et al [2008]. We denote our model using all 22 features by *rf22* (*rf* stands for Random Forests). Similarly, *rf7* consists of our top 7 features. The datasets generated for the experiments contain only trust and lack of trust statements, which allows our approach to be directly compared against the other methods. The 2-million data-set preserves the original distributions for trust and lack of trust statements. This provides a stratified experimental setup. Lastly, there exists review write-rate relationships between the trustor and trustee candidates in the dataset (i.e. the trustor gave a rating to one of the reviews produced by the trustee). This allows the Antecedent Framework model to score the candidate pairs from the data.[1] The top seven features of *top7* include features 1,2,4,5,6,8, and 9. The data-set generated for the first two experiments includes 400,000 trust statements and 1,600,000 randomly selected lack of trust statements.

We applied the Random Forests Classifier from Weka (cf. Hall et al [2009]) with the number of trees equal to 30, and the maximum depth of each tree equal to 100 in order to build the models for each set of features. The J48 algorithm is used to grow a single tree. The models were evaluated using a ten-fold cross-validation.

We also compare all three approaches using Random Forests and Support Vector Machines trained on the smaller dataset containing 1000 trusts and 1000 lack of trust statements. There is no comparison between *SVM* and *RF* on the 2-million dataset, as training the SVM classifier became impractical due to time constraints.

4.3 Random Forest and Support Vector Machine comparision

Figures 3 [left] and 4 compare the results for the models constructed by Random Forest and Support Vector Machines on the 2000-instance dataset. In overall, RFs outperforms SVM on this dataset showing higher scores for precision, recall, F-measure and ROC Area. Using SVM reduces the scores for both models. *ant*8 and *top*7 appear to be a bit more stable than our model when using different classifiers. The scores for the two are only slightly worse than the ones received for our model, when using the SVM classifier. Our model gives the best results in precision among all models for both classifiers: 0.8 and 0.73. The recall scores for both classifiers are somewhat contradictory. Random Forest yields a better recall of 0.8 for our model, whereas SVM improves the recall for *svm_ant*8 up to 0.76 outperforming our model by 3%. SVM gives tighter results in terms of F-measure (0.737, 0.715, and 0.126) and ROC Area (0.737, 0.696, and 0.515) between all approaches, with our model performing better than the other two.

4.4 Random Forests models

The results of using the Random Forests classifier on the two-million instance dataset are summarized on Figures 3 [right] and 5. In general, all metrics show better scores for *rf22* and *rf7* over the other two models.

[1] The features of *ant8* were computed with $\mu = 5$ and $\alpha = 0.1$.

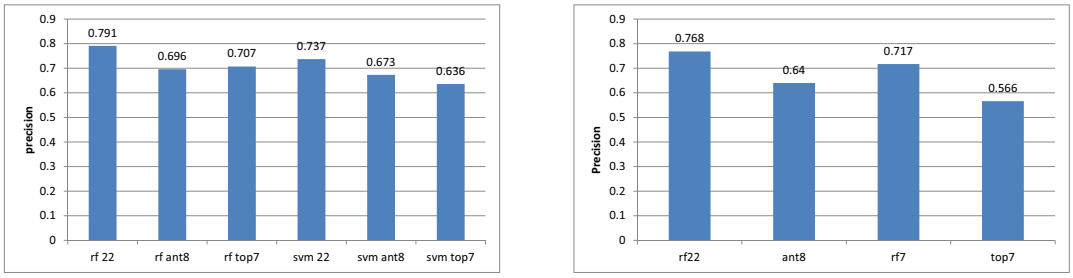

Fig. 3 Precision scores on the 2000 and 2,000,000 -instance datasets, respectively

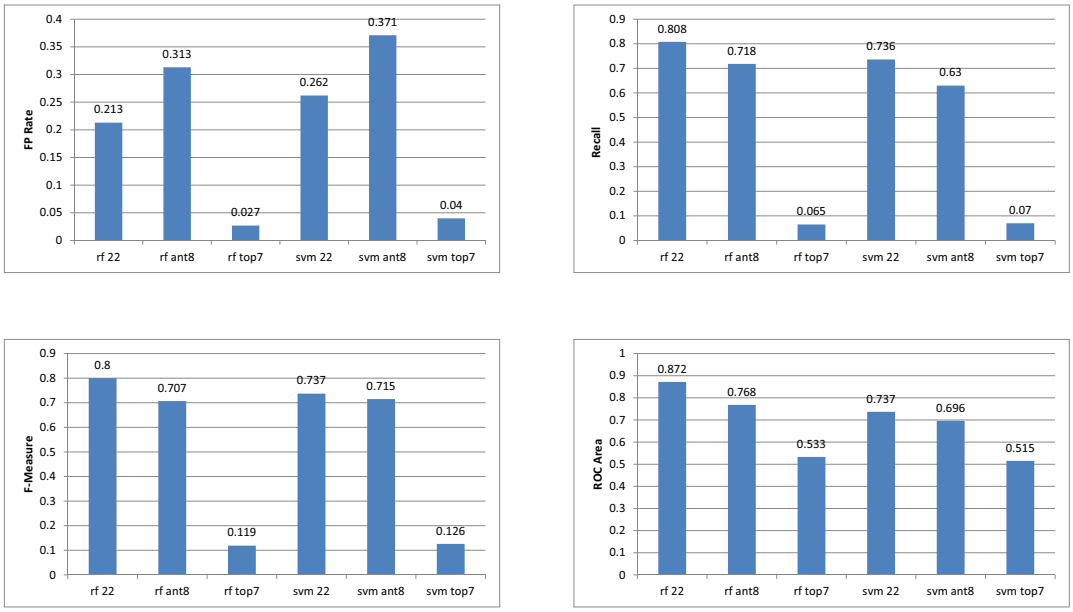

Fig. 4 FP Rate, Recall, F-Measure, ROC Area for RFs and SVMs on the 2000-instance dataset

rf22 shows significant improvements of 5% and 20% in precision, over *ant8* (0.64) and *top7* (0.57), respectively.

rf22 yields the best FP rate of 0.048 followed by *ant8* (0.056).

The recall metrics for *rf22* and *rf7* is about 20% greater than *ant8*.

F-measure reflects the combination of precision and recall scores by showing a 4% improvement for our model (0.7) over *ant8* (0.66).

Lastly, ROC Area shows that all classifiers perform better than random prediction. *rf22* and *rf7* show a 10% improvement over *ant8* for this metrics.

	Experiments		
	$score_1$	$score_2$	$score_3$
Precision	0.5707	0.4383	0.5660
Recall	0.3439	0.1573	0.3460
F-measure	0.4292	0.2315	0.4295
ROC Area	0.6084	0.5867	0.6090

Table 2 Precision, Recall, F-measure and ROC Area for the PTP model

5 Conclusions and Future Work

We proposed a comprehensive set of features to compute in order to perform accurate trust prediction. Our features capture user-similarity factors and rater-reviewer interactions. Then, we experiemented with employing effective classifiers using our features and those of com-

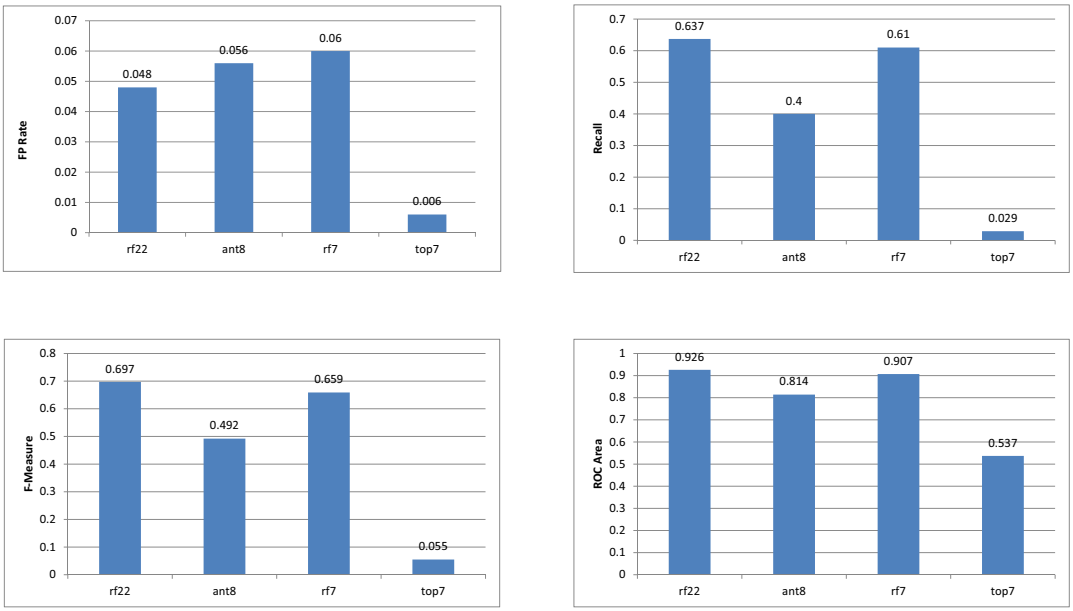

Fig. 5 FP Rate, Recall, F-Measure ROC Area for RFs on the 2,000,000-instance dataset

peting approaches. In general, our features increase the performance of classification algorithms by 5-20% compared to the previous approaches.

There are a couple of intuitions which might be worth developing. For example, we typically value the opinions of our family members and closest friends more than the ones of buddies or acquaintances. One could include this intuition to come up with features such as the number of the trustees of u (the users u trusts) who trust reviewer v, number of implicit ratings by the trustees of u given to the reviews written by v, number of implicit ratings by the trustees of u (who trust v as well) given to the reviews written by v, average rating by the trustees of u given to the reviews written by v, and average rating by the trustees of u (who trust v as well) given to the reviews written by v could be computed and employed for trust prediction.

The trust graph model could be extended to include different types of nodes or arcs to reconcile user similarity and trust information. Augmenting such trust propagation algorithms to use weights derived from Rater-Reviewer or User-Similarity information might result in performance gains and would be a promising direction for further research. Finally, the time factor can to be taken into consideration while developing or extending trust prediction algorithms. For example, traditional trust propagation algorithms treat a node's neighbors equally. Yet the initial neighbors are typically family members and close friends, whereas the most recent ones might be simple acquaintances. On the other hand, some older links might get weaken over time as compared to more recent ones.

In general, there are quite a few ideas and intuitions beyond those we captured in this paper that make sense in practice and which can be incorporated in a trust prediction problem. Therefore, we believe the landscape of trust prediction is quite rich and with a lot of potential for further results.

References

Borzymek P, Sydow M, Wierzbicki A (2009) Enriching trust prediction model in social network with user rating similarity. In: CASoN, pp 40–47

Chowdhury M, Thomo A, Wadge WW (2009) Trust-based infinitesimals for enhanced collaborative filtering. In: COMAD

Golbeck J (2005) Computing and applying trust in web-based social networks. (phd thesis)

Guha RV, Kumar R, Raghavan P, Tomkins A (2004) Propagation of trust and distrust. In: WWW, pp 403–412

Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The weka data mining software: an update. SIGKDD Explor Newsl 11(1):10–18

Jamali M, Ester M (2009) *TrustWalker*: a random walk model for combining trust-based and item-based recommendation. In: KDD, pp 397–406

Koren Y (2010) Collaborative filtering with temporal dynamics. Commun ACM 53:89–97

Kuter U, Golbeck J (2007) Sunny: A new algorithm for trust inference in social networks using probabilistic confidence models. In: AAAI'07, pp 1377–1382

Liu H, Lim EP, Lauw HW, Le MT, Sun A, Srivastava J, Kim YA (2008) Predicting trusts among users of online communities: an epinions case study. In: ACM Conference on Electronic Commerce, pp 310–319

Ma N, Lim EP, Nguyen VA, Sun A, Liu H (2009) Trust relationship prediction using online product review data. In: CIKM-CNIKM, pp 47–54

Massa P, Avesani P (2005) Controversial users demand local trust metrics: An experimental study on epinions.com community. In: AAAI, pp 121–126

Massa P, Avesani P (2009) Trust metrics in recommender systems. In: Computing with Social Trust, pp 259–285

Nguyen VA, Lim EP, Jiang J, Sun A (2009) To trust or not to trust? predicting online trusts using trust antecedent framework. In: ICDM, pp 896 –901

Noor TH, Sheng QZ (2011) Credibility-based trust management for services in cloud environments. In: ICSOC, pp 328–343

Sherchan W, Nepal S, Bouguettaya A (2011) A trust prediction model for service web. In: TrustCom, pp 258–265

Sinclaire J, Simon J, Wilkes R (2010) A prediction model for initial trust formation in electronic commerce. In: International Business Research, pp 17–27

Skopik F, Schall D, Dustdar S (2009) Start trusting strangers? bootstrapping and prediction of trust. In: WISE, pp 275–289