Predicting Trusts among Users of Online Communities – an Epinions Case Study

Haifeng Liu, Ee-Peng Lim
Hady W. Lauw
Minh-Tam Le, Aixin Sun
Nanyang Technological
University, Singapore
{aseplim,axsun}@ntu.edu.sg

Jaideep Srivastava
University of Minnesota
Minneapolis, MN, USA
srivastava@cs.umn.edu

Young Ae Kim
KAIST Business School
87 Hoegiro Dongdaemoon-gu
Seoul, Korea
kya1030@business.kaist.ac.kr

ABSTRACT

Trust between a pair of users is an important piece of information for users in an online community (such as electronic commerce websites and product review websites) where users may rely on trust information to make decisions. In this paper, we address the problem of predicting whether a user trusts another user. Most prior work infers unknown trust ratings from known trust ratings. The effectiveness of this approach depends on the connectivity of the known web of trust and can be quite poor when the connectivity is very sparse which is often the case in an online community. In this paper, we therefore propose a classification approach to address the trust prediction problem. We develop a taxonomy to obtain an extensive set of relevant features derived from user attributes and user interactions in an online community. As a test case, we apply the approach to data collected from Epinions, a large product review community that supports various types of interactions as well as a web of trust that can be used for training and evaluation. Empirical results show that the trust among users can be effectively predicted using pre-trained classifiers.

Categories and Subject Descriptors

H.3.5 [**Online Information Services**]: Web-based services; H.2.8 [**Database Applications**]: Data mining

General Terms

Experimentation

Keywords

Trust prediction, User interaction, Online community

1. INTRODUCTION

Trust between a pair of users is an important aspect of decision making for Internet users, and particularly for users of

an online community (such as electronic commerce websites and product review websites) where one user may rely on information provided by other users to make decision. A seller trusted by a buyer in an e-commerce website would have a significant advantage against other sellers in case the product quality cannot be verified in advance. Likewise, a user in a product review community is likely to refer to product reviews provided by his/her trusted reviewers. Though trust plays a crucial role in online communities, it is often hard to assess the trustworthiness between two users without their self-reporting.

There is a rapidly growing literature on inferring unknown trust ratings from known trust ratings among users (refer to Section 2). Most work on trust inference rely on a basic web of trust represented by a trust matrix which indicates whom a user trust or how much a user trusts others. However, the assumption that the existence of a basic web of trust is too strong for many online communities. In reality, either there is no way to know a web of trust, or the available web of trust is too sparse [6, 10]. We therefore aim to infer the trust relationship between two users solely based on their individual actions and interactions in an online community. In other words, we propose to build up a web of trust based on users' behaviors.

We observed that a user trusts another user either because of the latter's good reputation or because there have been good personal interactions between the two users. Therefore, we propose a supervised learning approach that automatically predicts trust between a pair of users using evidence derived from actions of individual users (*user factors*) as well as from interactions between pairs of users (*interaction factors*). Based on these factors, we derive the corresponding features to train classifiers that predict trust between pairs of users. As a case study, we applied our approach on Epinions [1], a large product review community that supports various types of interactions as well as a web of trust that can be used for training and evaluation. Our empirical results showed that (i) our trained classifiers can achieve satisfactory accuracy, and (ii) interaction factors have greater impact on trust decisions than user factors.

To the best of our knowledge, this is the first study of predicting pairwise user trust using a classification approach. We summarize our technical contributions below.

1. We developed a taxonomy to systematically organize an extensive set of features for predicting trust between a pair of Epinions users. This taxonomy is general enough to be adopted in other online communities,

and the features are powerful enough to achieve satisfactory prediction accuracy.

2. We conducted experiments with a few classification methods to evaluate their performance in predicting trust against the real-life web of trust. The best methods were reported.

3. We evaluated each feature and obtained top ranked features on deciding pairwise user trust.

4. We found that interactions between two users play a more important role than individual user actions in deciding pairwise user trust.

The rest of the paper is organized as follows. Section 2 presents some related work and Section 3 introduces notations and definitions used throughout the paper. We then develop a taxonomy of trust factors which are used to train classifiers in Section 4. Section 5 presents the experiment data collected from Epinions and experimental setup. We present and discuss the experiment results in Section 6. We finally draw conclusions in Section 7.

2. RELATED WORK

Computational trust models have recently been drawing the attention of researchers as more online service provision and communities are available to Internet users. The common ways of determining trust are through *reputation* [3, 8]. A user can gauge the reputation of another user based on the past interactions between the former and the latter (personal experience) as well as the interactions between the latter and other users (recommendations). In the latter case (to which most Internet users belong), one can use referral-based trust to compute trust in the absence of first-hand knowledge. Inferring trust from known trust heavily relies on a pre-known web of trust that allow users to express trust of other users. Guha et al. [6] proposes a trust propagation model to predict trust between two users without prior interaction through such a Web of trust. Other work on propagating trust through web of trust include [5, 9, 12, 13, 14].

The previous work requires the explicit trust ratings from users which are not always available from the online communities. Even when such information is available, the web of trust is often too sparse to infer the trust of two users without direct trust connectivity. In this paper, we thus propose a supervised learning approach to address the issue. Another work to infer trust has been reported in [10]. However, this work is to build a trust model based on users' reputation and affinity which is different from the approach taken in this paper. We also noted that the similar learning approach has been successfully applied in other problems of online communities, such as finding high-quality content in a web-based question answering community [2] and link prediction problem [7].

There are also a few other efforts to evaluate trust without a known trust network among users in the areas of P2P application, e-commerce and multi-agent systems [4, 11, 15, 16]. However, the trust models in these efforts require incorporating specific trust factors into a trust function, and are difficult to be applied to application domains where such a trust function may not exist.
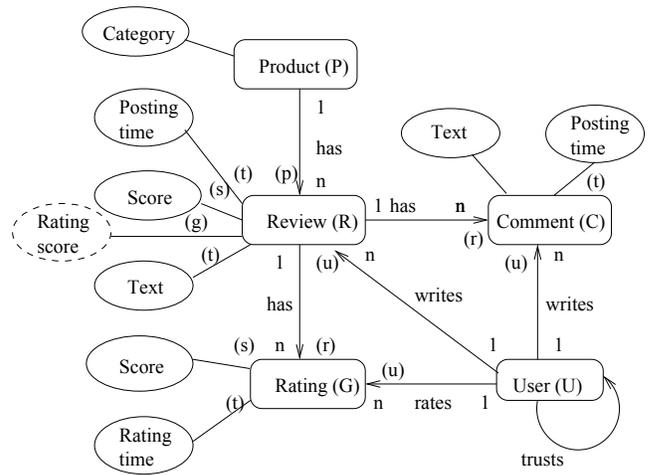

Figure 1: ER diagram of Epinions product review community.[1]

3. USERS, OBJECTS, AND INTERACTIONS IN EPINIONS

Epinions is an online product review community where anybody from the public can sign up as a member. Figure 1 shows the entity-relationship diagram of Epinions product review community. This online product review community can be represented as a 5-tuple: $E = <U, P, R, G, C>$ where U is the set of registered users/members, P is the set of listed products belonging to different categories, R is the set of reviews written by the users, G is the set of ratings given to the reviews, and C is the set of comments written by the users. All reviews, ratings, comments and products are *objects* of the community. The product reviews written by users consist of prose text and quantitative ratings from 1 to 5 stars. In this work, we identify three roles of a user. A user can be:

(a) a *review writer* who writes a review article for a product where the review assigns a numeric score between 0 and 1 to the product(With a linear mapping between scores and stars, 0 corresponds to 1 star whereas 1 corresponds to 5 stars),

(b) a *review rater* who rates a review with a score ranging from 0 to 1. In Epinions, users rate reviews as "Off Topic", "Not Helpful", "Somewhat Helpful", "Helpful", "Very Helpful" or "Most Helpful" which determine how prominently the review will be placed as well as to giving the reviewer a higher status. In this work, with a linear mapping a rating score 0 corresponds to "Off Topic" whereas 1 corresponds to "Most Helpful", and

(c) a *review commenter* who comments a review or a review comment. While a rater can only rate a review once, a commenter can post more than one comment to a review.

In addition, a web of trust is available for users to express who they trust. In this work, we refer to a *trustor* as a user who trusts or does not trust another user whereas we refer

---

[1](<uppercase letter>): name of entity set; (<lowercase letter>): name of attribute.

to a *trustee* as the user who is trusted by some other user(s). An user is a *candidate trustee* if he or she is to be considered as a trustee of some trustor.

We identified the following attributes of interest for the objects:

- For a review r, its attributes include $p(r)$: the product it reviews, $s(r)$: the review score assigned to the product, $t(r)$: the time of posting the review, $u(r)$: the user who writes r, $g(r)$: the overall rating score that is derived from all ratings received by r, and the text content[1] of r.

- For a rating g, its attributes include $r(g)$: the review it rates, $s(g)$: the rating score given to the review, $t(g)$: the time of rating, and $u(g)$: the user who rates.

- For a comment c, its attributes include $r(c)$: the review it comments, $t(c)$: the time of commenting, $u(c)$: the user who comments, and the text content of c.

- For a product p, the only attribute we used in this work is its category.

Relationships among Objects

In terms of relationships among objects, we define *competing reviews*, *competing ratings*, *competing comments*, and *sibling comments* as follows:

DEFINITION 1. **Competing Reviews** *Two reviews r_i and r_j are competing reviews if they review the same product, $p(r_i) = p(r_j)$.*

DEFINITION 2. **Competing Ratings** *Two ratings g_i and g_j are competing ratings if they rate the same review, $r(g_i) = r(g_j)$.*

DEFINITION 3. **Competing Comments** *Two comments c_i and c_j are competing comments if they comment the same review and are given by different users, i.e., $r(c_i) = r(c_j)$ and $u(c_i) \neq u(c_j)$.*

DEFINITION 4. **Sibling Comments** *Two comments c_i and c_j are sibling comments if they comment the same review and are given by the same user, i.e., $r(c_i) = r(c_j)$ and $u(c_i) = u(c_j)$.*

We define the agreement between a review and its competing reviews, and the agreement between a rating and its competing ratings below.

DEFINITION 5. **Agreement between a Review and its Competing Reviews** *Let $\Gamma(r)$ be the set of competing reviews with a review r. The agreement between r and all its competing reviews is*

$$\theta(r) = 1 - \frac{\sum_{r_i \in \Gamma(r)} |s(r) - s(r_i)|}{|\Gamma(r)|}$$

DEFINITION 6. **Agreement between a Rating and its Competing Ratings** *Let $\Delta(g)$ be the set of competing ratings with a rating g. The agreement between g and all its competing ratings is*

$$\theta(g) = 1 - \frac{\sum_{g_i \in \Delta(g)} |s(g) - s(g_i)|}{|\Delta(g)|}$$

[1]Note that, we do not do any content analysis in this work. The only explicit attribute we derived from content is the length of content

Types of User Interactions

The users in the community may interact with each other through the various objects. We identify a few user interactions based on the diagram in Figure 1 with the following definitions:

DEFINITION 7. **Connection** *Two users are connected through an object if both users have performed some interaction on the object.*

DEFINITION 8. **Write-Rate (WR) Connection** *Given two users $u_i, u_j \in U$, if u_i writes a review $r \in R$ and u_j rates r, then an WR connection is formed between u_i and u_j, and denoted as $wr(u_i, u_j, r)$.*

DEFINITION 9. **Rate-Rate (RR) Connection** *Given two users $u_i, u_j \in U$, if after u_i rates a review $r \in R$, u_j rates r as well, then an RR connection is formed between u_i and u_j, and denoted as $rr(u_i, u_j, r)$.*

DEFINITION 10. **Write-Write (WW) Connection** *Given two users $u_i, u_j \in U$, if after u_i writes a review $r_i \in R$ about a product $p \in P$, u_j writes another review $r_j \in R$ about p as well, then an WW connection is formed between u_i and u_j, and denoted as $ww(u_i, u_j, p)$.*

DEFINITION 11. **Write-Comment (WC) Connection** *Given two users $u_i, u_j \in U$, if u_i writes a review $r \in R$ and u_j comments on r, then an WC connection is formed between u_i and u_j, and denoted as $wc(u_i, u_j, r)$.*

Note that u_j's multiple comments on u_i's one review r will be represented by one WC connection only: $wc(u_i, u_j, r)$.

DEFINITION 12. **Comment-Comment (CC) Connection** *Given two users $u_i, u_j \in U$, if after u_i comments on a review $r \in R$, u_j comments on r or u_i's comment, then an CC connection is formed between u_i and u_j, and denoted as $cc(u_i, u_j, r)$.*

Besides the above direct connections, there are some indirect interactions between two users. For example, it is possible that one may write his own review on a product after seeing another user's rating on a review on the same product. This suggests a Rate-Write (RW) connection between two users. However, as this is hard to determine (as it is hard to tell whether a review is written because the writer saw a rating on another competing review or not) and has a weaker link with the trustworthiness between users, we do not consider it in this work. Similarly, we do not include possible Comment-Write (CW), Comment-Rate (CR) and Rate-Comment (RC) connections.

4. TAXONOMY OF TRUST FACTORS

In this section, we present a taxonomy of trust factors that will be used by learning algorithms as input features to train binary classifiers. With this taxonomy, one can systematically enumerate the trust factors that can be derived from the product review data. The taxonomy divides trust factors into two main categories, namely *user factors* and *interaction factor*. The former refers to features associated with a given user who can be a trustor or trustee as shown in Figure 2. The latter refers to features associated with the interaction that occurs between a pair of users in the trustor-trustee roles as shown in Figure 3.
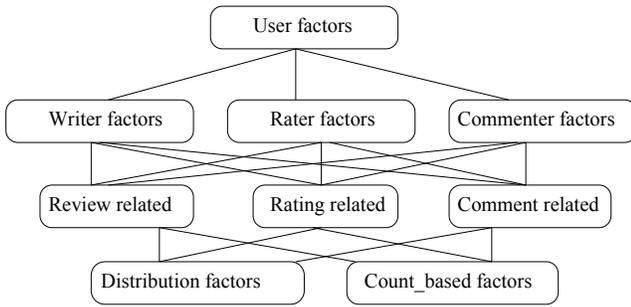
Figure 2: An taxonomy of user factors

4.1 Taxonomy of User Factors

Figure 2 depicts a structured taxonomy of user factors. User factors are those factors pertaining to the publicly known actions of the user as a review writer, a review rater or a review commenter. To predict whether user u_j trusts user u_i, the user factors of u_i and the user factors of u_j can be determined. The user factors of u_i (or u_j) are derived from the actions (e.g., writing reviews, giving comments, etc.) that u_i (or u_j) has performed. While these actions are not necessarily related to u_j (or u_i), they influence how trust may be built up between u_i and u_j. For example, if u_i consistently writes a lot of good quality reviews (recognized as "top reviewers" in Epinions), then u_j may trust u_i. on the other hand, if u_i is inactive in the community, it is also possible that u_j will not put u_i in his trustee list.

Regardless of the role of a user, the factors are divided into three groups: *review related*, *rating related*, and *comment related*. For all the factors in each group above, they are further divided into two sub-groups: *distribution factors* and *count-based factors*. Distribution factors are those factors that can be measured by the statistics metrics such as average and standard deviation, while count-based factors are those which are related to counting a specific set of objects. We then quantify each factor with two features. An *absolute feature* is the absolute value of a factor while a *normalized feature* is the Min-Max normalized value of the factor (relative to all users with the same role). For example, if an user u_i writes 10 reviews in total, and the minimum and maximum number of reviews written by a writer are 1 and 100 respectively, then for the trust factor "number of review posted", the absolute feature value is 10 while the normalized feature value is $10 - 1/100 - 1 = 0.09$. These two groups of features derived from all trust factors are taken as input features for classifiers in Section 5.

4.1.1 Review related

This group of factors considers the aggregated attributes associated with a set of reviews written by a user (in trustor or trustee role) without considering their received ratings and comments.

Distribution factors. Currently, the following attributes associated with a review are considered:

- *Review score to its product.* The attribute reflects how a review writer directly evaluates a product.

- *Review content length.* As we do not analyze the content of review, we assume that review length can serve as a proxy towards review quality.

- *Number of competing reviews.* This may give us the impression of a user's competence as a writer which is related to the trustworthiness of the writer.

- *Number of competing reviews posted before it.* This may also indicate the competence of a user as a writer. We did not consider the number of competing reviews after because that is very much beyond the choice of the user as a review writer.

- *Agreement with its all competing reviews.* This may indicate the agreeability of a writer as we doubt that it is also related to trustworthiness.

- *Agreement with the competing reviews posted before it.* This is another factor linked to the agreeability of a writer.

- *Overall product score given by all competing reviews.* Defined as the average of product scores given by all competing reviews, this reflects the quality of a product reviewed by a writer, which allows us to understand the product preference of a writer.

As each user may write multiple reviews, we obtain the trust factors by aggregating the above attribute values of his/her reviews using five summary measures: minimum, maximum, median, average, and standard deviation of attribute values.

Count-based factors. We also consider the following count-based factors:

- *Number of reviews posted.* It directly indicates the review-writing experience of a writer.

- *Review frequency.* This refers to reviews posted per time unit. We consider it as a count-based factor as it is derived from the number of reviews posted.

- *Number of product categories involved.* This indicates the knowledge width of a writer.

- *Number of express reviews.* There are two types of reviews in Epinions: *normal reviews* and *express reviews*. Express reviews are short, concise reviews that sum up a reviewer's experience quickly. They are at least 20 words and less than 200 words. Normal reviews contain more than 200 words.

- *Number of first reviews.* A first review is the one that is posted earliest among all competing reviews on the same product. This indicates how quick the writer is to respond to the new product in the market.

- *Number of the reviews with their overall received scores higher than a specified threshold.* In our experiments, we set the threshold as 0.8 which allows us to select the reviews marked with "Very Helpful" and "Most Helpful".

- *Number of the reviews with their overall received scores lower than a specified threshold.* In our experiments, we set the threshold as 0.2 which allows us to select the reviews marked with "Off Topic" and "Not Helpful".

- *Number of the reviews with their product scores higher than a given threshold.* We set the threshold as 0.8 in the experiments which allows us to select 4 or 5 stars products.

- *Number of the reviews with their product scores lower than a given threshold.* We set the threshold as 0.2 in the experiments which allows us to select one star products.

4.1.2 Rating related

The rating related factors consider all attributes associated with a set of ratings given by a user (in trustor or trustee role) without considering the reviews he/she rates and the comments he/she gives to those reviews. Similar to the review related factors, they are grouped by distribution factors and count-based factors.

Distribution factors.

- *Rating score given to its targeted review.* This attribute directly reflects how a rater evaluate a review.

- *Number of competing ratings.* This (with the next attribute) may indicate the competence of user as a rater.

- *Number of competing ratings given before it.*

- *Agreement with its competing ratings.* This (with the next attribute) may indicate the agreeability of user as a rater.

- *Agreement with the competing ratings given before it.*

Each attribute above is associated with the same five summary measures used for review rated factors in Section 4.1.1.

Count-based factors.

- *Number of ratings given.* This directly reflects the experience of the user as rater.

- *Rating frequency.* This refers to ratings given per time unit.

- *Number of first ratings.* First rating is the one that is given earliest among all competing ratings on the same review. This may indicate how quick for a rater to respond to newly posted reviews.

- *Number of the ratings that are higher than a specified score threshold.* We set the threshold as 0.8 in the experiments which allows us to select the "Very Helpful" and "Most Helpful" ratings.

- *Number of the ratings that are lower than a specified score threshold.* We set the threshold as 0.2 in the experiments which allows us to select the "Off Topic" and "Not Helpful" ratings.

4.1.3 Comment related

The comment related factors consider all attributes associated with a set of comments given by a user (in trustor or trustee role) without considering the reviews they comment and the ratings received by those comments. Like the review and rating related factors, they are grouped by distribution factors and count-based factors.

Distribution factors.

- *Comment length.* This indirectly reflects the comment quality.

- *Number of sibling comments.* This reflects the effort of the user as a commenter on one review.

- *Number of competing comments.* This (with the next two attributes) may indicate the competence of user as a commenter.

- *Number of competing commenters.*

- *Number of competing comments posted before it.*

Each attribute above is associated with the same five summary measures used for review rated factors in Section 4.1.1.

Count-based factors.

- *Number of comments given.* This directly reflects the experience of the user as a commenter.

- *Commenting frequency.* This refers to comments given per time unit.

- *Number of first comments.* First comment is the one that is given earliest among all comments including its sibling and competing comments on the same review.

4.1.4 Generating user factors

As shown in Figure 1, a user can contribute reviews, ratings, and comments. Depending on the type of contribution, the above three groups of factors can be applied to different sets of objects as described below to obtain *writer factors*, *rate factors* and *commenter factors*.

- For generating writer factors, review related factors are meant for all reviews written by the user as a writer, rating and comment related factors are meant for all ratings and comments that are received by the reviews written by the user.

- For generating rater factors, review related factors are meant for all reviews to which the user as a rater has given ratings, comment related factors are meant for all comments that are given to the reviews to which the user has given ratings, and rating related factors are meant for all ratings that he/she has given.

- For generating commenter factors, review related factors are meant for all reviews to which the user as a commenter has given comments, comment related factors are meant for all comments that the user has given, and rating related factors are meant for all ratings that are given to the reviews he/she has commented.

4.2 Taxonomy of Interaction Factors

The interaction factors are the factors that directly influence a user's decision based on his/her personal connections with another user. Figure 3 depicts a structured taxonomy of interaction factors. To decide whether u_j trusts u_i based on his/her personal experience with u_i, the following trust factors are derived from all interactions between them, i.e., $WR_{u_i u_j}$, $WW_{u_i u_j}$, $RR_{u_i u_j}$, $WC_{u_i u_j}$, and $CC_{u_i u_j}$. As each connection involves two users, the factors are categorized into three groups: localized (trustor) user factors, localized (trustee) user factors, and a connection specific temporal factor. Localized user factors are all user factors (as in Section 4.1) confined by the set of a particular type of user connections. Depending on the role of user involved with the connections, *localized user factors* refer to either localized writer factors, or localized rater factors, or localized
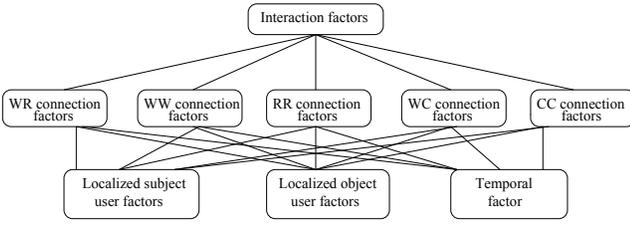
Figure 3: An taxonomy of interaction factors

commenter factors. *Temporal factor* for a connection is the time difference between two user's respective actions which form the connection. For example, for $WR_{u_i u_j}$ connections, there would be localized writer factors (related to u_i), localized rater factors (related to u_j), and a write-rate connection specific temporal factor that is the response time for the rater to rate the writer's review. Again, each factor would be associated with one absolute feature and one normalized feature (A temporal factor would have two normalized features).

4.2.1 WR Connection Factors

Localized user factors for WR. In the context of $WR_{u_i u_j}$ connections, we perform the localization of user factors as follows:

- Let R_{ij} denote the set of reviews written by u_i and rated by u_j. *Localized review factors for writer u_i* are the review related factors derived from R_{ij}.

- *Localized rating factors for u_i* are the rating related factors derived from the set G_{ij} of ratings given to the set of reviews written by u_i and rated by u_j. That is, $G_{ij} = \{g | r(g) = x \text{ where } x \in R_{ij}\}$.

- *Localized comment factors for u_i* are the comment related factors derived from the set C_{ij} of comments given to the set of reviews written by u_i and rated by u_j. That is, $C_{ij} = \{c | r(c) = x \text{ where } x \in R_{ij}\}$.

- *Localized review factors for rater u_j* are the review related factors derived from R_{ij}.

- *Localized rating factors for u_j* are the rating related factors derived from the set G'_{ij} of u_j's ratings given to the set of reviews written by u_i. That is, $G'_{ij} = \{g | u(g) = u_j \text{ and } r(g) = x \text{ where } x \in R_{ij}\}$.

- *Localized comment factors for u_j* are the comment related factors computed from C_{ij}.

As we restrict that one rater can only give one rating to a review, the absolute feature set of localized review factors for u_j is the same with the absolute feature set of localized review factors for u_i. And the absolute feature set of localized comment factors for u_j is the same with the absolute feature set of localized comment factors for u_i. However, normalized feature sets for u_i and u_j are different. That is, the values of normalized features for u_i are relative to all writers whose reviews have been rated by u_j while the values of normalized features for u_j are relative to all raters who have rated u_i's reviews. We thus exclude the absolute feature sets of localized review and comment factors for u_j. Similarly, the absolute feature "number of reviews" in the

set of localized review factors for u_i is the same with the absolute feature "number of ratings" in the set of localized rating factors for u_j as both refer to the number of $WR_{u_i u_j}$ connections. We also exclude the absolute feature "number of ratings" of u_j.

Both absolute and normalized feature sets of localized rating factors are different between u_i and u_j as they are derived from two rating sets G_{ij} and G'_{ij} respectively.

WR specific temporal factor. Currently, we identify one specific temporal factor for WR connections: *the response time for u_j to rate u_i's reviews* which is defined as the difference between their posting dates. As there can be multiple WR connections between two users, five summary measures are associated with this factor as well: minimum, maximum, median, average, and standard deviation. For each measure, besides the absolute feature, we also derive two normalized features respectively relative to all users who rate u_i's reviews and all users whose reviews are rated by u_j. This results in 15 WR specific temporal features.

4.2.2 WW Connection Factors

Localized user factors for WW. In the context of $WW_{u_i u_j}$ connections, we perform localization as follows:

- *Localized review factors for writer u_i* are the review related factors computed from the set R_{ij} of reviews written by u_i and followed by u_j's reviews.

- *Localized rating factors for u_i* are the rating related factors computed from the set G of ratings given to the set of reviews written by u_i and followed by u_j. That is, $G_{ij} = \{g | r(g) = x \text{ where } x \in R_{ij}\}$.

- *Localized comment factors for u_i* are the comment related factors computed from the set C_{ij} of comments given to the set of reviews written by u_i and followed by u_j's reviews. That is, $C_{ij} = \{c | r(c) = x \text{ where } x \in R_{ij}\}$.

- *Localized review factors for writer u_j* are the review related factors computed from the set R'_{ij} of reviews written by u_j and posted after u_i's reviews.

- *Localized rating factors for u_j* are the rating related factors computed from the set G'_{ij} of ratings given to the set of reviews written by u_j and posted after u_i's reviews. That is, $G'_{ij} = \{g | r(g) = x \text{ where } x \in R'_{ij}\}$.

- *Localized comment factors for u_j* are the comment related factors computed from the set C'_{ij} of comments given to the set of reviews written by u_j and posted after u_i's reviews. That is, $C'_{ij} = \{c | r(c) = x \text{ where } x \in R'_{ij}\}$.

Clearly, the review factor "number of first reviews" is not applicable to u_j. In addition, the absolute features of "number of reviews posted", "number of competing reviews", "overall product score given by all competing reviews", and "number of product categories" are overlapped with the corresponding absolute features of u_i's review factors, and they are excluded from the u_j's review factor absolute feature set.

WW specific temporal factor. Similarly to the WR specific temporal factor, we have 15 WW specific features in terms of the response time for u_j to follow u_i's reviews.

4.2.3 RR Connection Factors

Localized user factors for RR. In the context of $RR_{u_i u_j}$ connections, we perform localization as follows:

- *Localized review factors for rater u_i* are the review related factors derived from the set R_{ij} of reviews rated by both u_i and u_j where u_i's rating is before u_j's rating.

- *Localized rating factors for u_i* are the rating related factors computed from the set G_{ij} of ratings followed by u_j's ratings.

- *Localized comment factors for u_i* are the comment related factors computed from the set C_{ij} of comments given to the set of reviews rated by both u_i and u_j where u_i's rating is before u_j's rating. That is, $C_{ij} = \{c | r(c) = x \text{ where } x \in R_{ij}\}$.

- *Localized review factors for rater u_j* are the review related factors derived from R_{ij}.

- *Localized rating factors for u_j* are the rating related factors computed from the set G'_{ij} of ratings that follow u_i's ratings.

- *Localized comment factors for u_j* are the comment related factors computed from C_{ij}.

Clearly, the absolute feature set of review factors for u_j is the same with the absolute feature set of review factors for u_i, and the absolute feature set of comment factors for u_j is the same with the absolute feature set of comment factors for u_i. We thus exclude u_j's absolute feature sets of review and comment factors. However, the corresponding normalized features sets for u_i and u_j are different. That is, the values of normalized features for u_i are relative to all raters whose ratings have been followed by u_j's ratings while the values of normalized features for u_j are relative to all raters who give ratings after u_i's ratings on the same reviews.

In addition, the rating factor "number of first ratings" is not applicable to u_j', and "number of competing ratings" and "number of ratings given" are overlapped with u_i's rating factors, we exclude them from the u_j's rating factor set.

RR specific temporal factor. Similarly to the WR and WW specific temporal factors, we have 15 RR specific features in terms of the response time for u_j to follow u_i's ratings.

We obtain the WC and CC connection factors in the similar way (WC is similar with WR and CC is similar with RR). Due to the space limitation, we do not describe their details.

5. EXPERIMENTAL SETUP

5.1 Epinions Data

We crawled Epinions product reviews of "Videos & DVDs" category with the ratings (but without comments) on July 22, 2007. Table 1 shows the data set statistics.

Figure 4(a) shows the distribution of the number of reviews written by each writer versus the count of writers, and Figure 4(b) shows the distribution of the number of ratings given by each rater versus the count of raters. Both figures are drawn with log scale on both axes, and exhibit similar

Table 1: Summary of Epinions Data Set.

Number of products	7,081
Number of reviews	80,348
Number of ratings	1,516,460
Number of users	42,503
Number of writers	24,821
Number of raters	27,460
Number of users who are both writers and raters	9,778

Table 2: Connected User Pairs and Trust Distribution.

| WR	WW	RR	User pairs	Positive pairs	Negative pairs
X	X	√	3,440,569(33.5%)	84,942(46.2%)	3,355,627(33.3%)
X	√	X	6,080,855(59.2%)	4,771(2.6%)	6,076,084(60.2%)
X	√	√	102,889(1.0%)	2,534(1.4%)	100,355(1.0%)
√	X	X	349,396(3.4%)	33,674(18.3%)	315,722(3.1%)
√	X	√	207,810(2.0%)	44,652(24.3%)	163,158(1.6%)
√	√	X	47,159(0.5%)	3,134(1.7%)	44,025(0.4%)
√	√	√	40,462(0.4%)	9,975(5.4%)	30,487(0.3%)
Total			10,269,140(100%)	183,682(100%)	10,085,458(100%)

characteristics. They approach a straight line on a log-log graph, which means they follow/approach the power law distribution. Since both axes are in log scale, these two figure mean that (a) most writers have very few reviews, and very few writers have extremely many reviews; (b) most raters give very few ratings, and very few raters have extremely many ratings. In fact, we found that the maximum number of reviews written by one writer is 412 and most (15,077) writers write only one review, while the maximum number of ratings given by one rater is 16,966 and lots of (10,531) raters give only one rating.

We scanned the above product review data set and identified 10,269,140 connected user pairs including three types of connections: WR, WW and RR (There are no WC and CC connections in the data set as we did not collected comments from Epinions). Note that the connections between a pair of users are directional. We also crawled the web of trust among the above 42,503 users. Figure 5 shows the distribution of number of trustees for each trustor. The figure is drawn with log scale on both axes, and exhibit similar characteristics as Figure 4(a) and 4(b). We observed that most trustors have very few trustees, and very few trustors have extremely many trustees, and that the maximum number of trustees for one trustor is 858 and most (3,489) trustors have only one trustee.

Table 2 summarizes the distribution of various connections among user pairs and the corresponding trust distribution (positive trust for a pair of users means that the trustor trusts the candidate trustee while negative trust means that the candidate trustee is not in the list of trustees of the trustor) where a "√" indicates that user pairs have the connection specified in the corresponding column head while a "X" indicates no such a connection. Initial observation from Table 2 shows that WR and RR connections have a relative strong power in affecting trust decisions.

5.2 Evaluation Methodology

Our goal is to train binary classifiers to predict if a user trusts another. A trustor in Epinions can do one of three things: (a) trusts another user (expressed in web of trust), (b) blocks (distrusts) another user, and (c) neither (a) or (b). However, due to restricted access of Epinions data, we
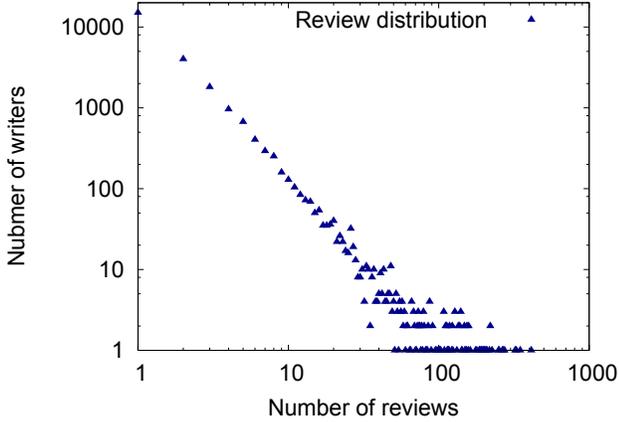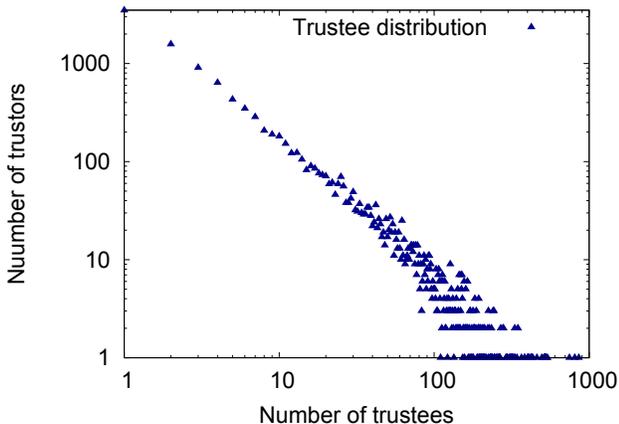
(a) Distribution of number of reviews v.s. number of writers.
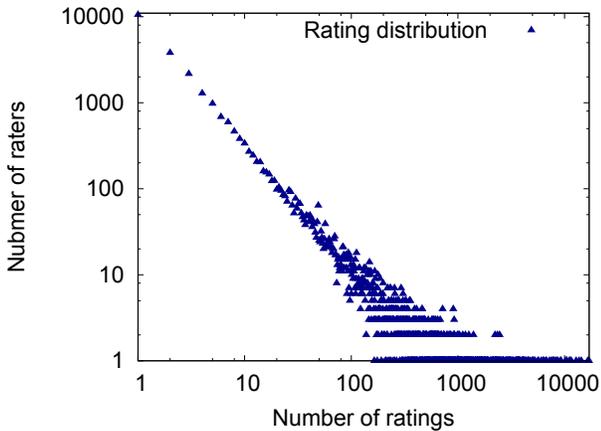
(b) Distribution of number of ratings v.s. number of raters.

Figure 4: Distribution of number of review-writer and number of rating-rater in our data.

Figure 5: Distribution of number of trustees v.s. number of trustors.

Table 3: Class and instance distribution in 5 folds

Fold	Trustors	Instances	Positive instances	Negative instances
1	467	8,360	1,963(23.5%)	6,397(76.5%)
2	466	7,832	2,092(26.7%)	5,740(73.3%)
3	466	7,971	1,989(25.0%)	5,982(75.0%)
4	466	8,061	2,043(25.3%)	6,018(74.7%)
5	466	8,238	1,888(22.9%)	6,350(77.1%)
Total	2,331	40,462	9,975(24.7%)	30,487(75.3%)

cannot differentiate between cases (b) and (c). Thus, in this work, a positive instance is a user pair where the trustor trusts the candidate trustee (i.e., case (a)) while a negative instance is a user pair where such a trust information is not available (e.g., cases (b) and (c)).

Due to limited computing resources (we used a desktop PC with 2GB memory and 3GHz CPU), we are unable to train the classifiers from the data set of all ten millions of user pairs. Instead, we chose a subset of 40,462 user pairs that have all three types of connections as the subset has the highest odds ratio (5.4% : 0.3%) between positive and negative classes which implies the highest discrimination power between positive and negative classes. As we have only a single product category and no comments in the data set, the trust factors summarized in Section 4 are not all available for the experiments. We managed to derive 576 user features (UF feature set) and 821 features for WR, WW and RR connections (IF feature set). In total, we have 1397 features (AF feature set). We use these three sets of features to respectively train classifies by employing different learning algorithms.

Performance of classifiers are validated through 5-fold cross validation. Figure 5 shows that there is a great variance for the number of trustees per trustor. To avoid bias towards either the trustors who easily trust a lot of people or the trustors who trust few people, we stratify the data set into 5 folds with the following procedure: first we find the users who are the first ones in the user pairs included in the date set, next sort this list in ascending order of the number of instances for each user. For the users having the same number of instances, we sort them again in ascending order of the number of trustees (positive instances) they have. Let u_i be the position of a user in the list and $k = (i \mod 5) + 1$, we then allocate all instances with the user as the first one into the kth fold. In this way, all folds share the similar number of instances and similar class distribution (see Table 3).

As we are interested in predicting trusted instances, we report the precision, recall and F-measure for the positive class. Besides, as the class distribution is not balanced (25% positive instances and 75% negative instances), and precision and recall are based on the whole set of instances returned by the classifiers and they do not account for the quality of ranking the hits, we also report the value of precision at 25% (25%-Precision) which is computed with the procedure below: First rank all instances in the ascending order of predicted scores, then predict the top 25% instances as positive. In this way, we focus on the accuracy of predicting positive class, and the values of precision and recall are the same.

6. RESULTS AND DISCUSSIONS

We experimented with several classification methods including decision tree, Naive Bayes (NB), logistic regression

Table 4: Performance of with different feature sets

Feature set	Method	Precision	Recall	F-value	25%-Precision
UF	NB	37.5%	68.9%	48.6%	47.6%
	SVM	67.9%	30.2%	41.8%	48.5%
IF	NB	44.0%	69.5%	53.9%	47.6%
	SVM	70.7%	36.0%	47.7%	58.3%
AF	NB	43.4%	69.7%	53.5%	46.5%
	SVM	72.0%	37.3%	49.1%	58.7%
Baseline		25%	25%	25%	25%

Table 5: Distribution of top 100 significant features.

Feature set	WR	WW	RR	User features
IF	78	0	22	NA
AF	60	0	14	26

and support vector machines with linear kernel and RBF kernel respectively. As the class distribution is known, we set a baseline classifier as one that randomly assign 25% instances as positive. Among the tested methods, the best performance was obtained by NB and SVM (RBF kernel) classifiers. We thus focus on the discussion of these two classifiers. Table 4 shows their classification performance using different feature sets compared to the baseline classifier.

The above results show that both NB and SVM classifiers outperform the baseline classifier, and the performance between NB and SVM are comparable. Both classifier performed about twice better than the baseline classifier in terms of F-value and 25%-Precision. Between SVM and NB, SVM achieved higher precision than NB while NB outperformed SVM regarding both recall and F-value. This is possibly because that NB and SVM take different score thresholds for separating positive instances from negative ones. NB takes an instance as positive when its predicted positive score is above 0.5 while SVM takes an instance as positive when its predicted score is above 0. In terms of 25%-Precision, SVM outperforms NB on all three feature sets. In addition, both classifiers achieved better performance when using interaction feature set compared to user feature set. For SVM, it achieved 14% and 20% better in terms of F-value and 25%-Precision respectively. For NB, it achieves 11% better in terms of F-value while there is no change on 25%-Precision values. However, when all features are employed by classifiers, there is no significant improvement of performance compared to that using interaction features only. For SVM, the F-values on IF and AF are 47.7% and 49.1% while the 25%-Precision on IF and AF are 58.7% and 58.3%. For NB, the performance on AF is even slightly worse than that on IF. This strongly implies that user features do not have good discriminating ability, and that interaction features do have deeper influence on making trust decisions.

Our another objective is to compare the features to judge their relative strength in predicting trust. As UF set does not have significant impact on predicting trust according to Table 4, we only evaluate individual features from IF and AF set. We did a chi-squared test for each set so as to rank the features in terms of their significance on inferring positive trust. Table 5 shows the distribution of the 100 most significant features among the different types of features in each set. From the table, we observed that WW features do not play major roles in inferring trust. This is consistent with the initial observation from Table 2. Besides, WR features are the most significant ones in deciding trusts.

We list the top 10 significant features for pairwise trust obtained from the IF set as follows:

1. WR feature: The absolute number of ratings with scores higher than 0.8 that are given to the writer by the rater.

2. WR feature: The absolute number of reviews with received overall rating scores higher than 0.8 that are written by the writer and rated by the rater.

3. WR feature: The absolute total number of reviews that are written by the writer and rated by the rater.

4. WR feature: The absolute number of total ratings that are given by the rater to the writer.

5. WR feature: The absolute number of ratings with scores higher than 0.8 that are given to the reviews that are written by the writer and ratted by the rater.

6. WR feature: The absolute total number of ratings that are given to the reviews that are written by the writer and ratted by the rater.

7. WR feature: The absolute number of reviews with product scores higher than 0.8 that are written by the writer and rated by the rater.

8. WR feature: The absolute number of first ratings that are given to the writer by the rater.

9. WR feature: The absolute total number of ratings that are given to the writer by the rater.

10. WR feature: The normalized total number of reviews that written by the writer and rated by the rater.

Interestingly, only WR features were included in the list. In fact, the top 20 list contains only WR features. The next frequently appeared connection features in the top list were RR features. The highest position in the top list achieved by the WW features was 268. Again, this indicates the significance of WR features and insignificance of WW features (probably because of the competence between two users writing reviews on the same product).

The top 10 significant features for pairwise trust obtained by chi-squared test from the AF set were as follows:

1. WR feature: The absolute total number of ratings that are given to the reviews that are written by the writer and ratted by the rater.

2. WR feature: The absolute number of ratings with scores higher than 0.8 that are given to the reviews that are written by the writer and ratted by the rater.

3. WR feature: The absolute number of ratings with scores higher than 0.8 that are given to the writer by the rater.

4. WR feature: The absolute number of first ratings that are given to the writer by the rater.

5. WR feature: The absolute total number of ratings that are given to the writer by the rater.

6. WR feature: The absolute total number of reviews that are written by the writer and rated by the rater.

7. WR feature: The absolute number of reviews with received overall rating scores higher than 0.8 that are written by the writer and rated by the rater.

8. WR feature: The absolute number of reviews with product scores higher than 0.8 that are written by the writer and rated by the rater.

9. WR feature: The normalized total number of ratings that are given to the writer by the rater.

10. WR feature: The normalize total number of reviews that rated by the rater.

The top 8 features from the AF set were also included in the top 10 list from the IF set though the order was slightly different. It implies that user features have no significant impact on trust decision. (In fact, there were no user features included in the top 50 features from the AF set).

7. CONCLUSIONS

Web of trust offers an important insight of the relationships among users in an online community. In this paper, we presented a classification approach to predict if a user trusts another user using features derived from his/her interactions with the latter as well as from the interactions with other users. The approach aims to predict missing trust information thereby enhancing the connectivity of a web of trust. As a case study, we apply the approach on the Epinions community. We first observed the user behaviors in the community, and identified an extensive set of trust features that may affect user's trust decisions from the perspectives of user's individual actions and interactions between a pair of users. The experiment results show that Naive Bayes and SVM classifiers using interaction features can perform better than those using user features only. We also found the Write Rate interaction features are more discriminatory than features based on other types of interactions. Although our approach is developed for online product review community, it is applicable to other online communities including e-commerce websites where sellers and buyers interact with one another. As part of our further research, we are interested in predicting the evolution of trust where the trust relationships among users change dynamically in time.

Acknowledgement

8. REFERENCES

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

[2] E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne. Finding high-quality content in social media with an application to community-based question answering. Technical Report YR-2007-005, Yahoo!, Sep. 2007.

[3] D. Artz and Y. Gil. A survey of trust in computer science and the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5:58–71, 2007.

[4] R. Ashri, S. D. Ramchurn, J. Sabater, M. Luck, and N. R. Jennings. Trust evaluation through relationship analysis. In *Proc. of 4th Int'l Joint Conference on Autonomous Agents and MultiAgent Systems*, pages 1005–1011, Utrecht, Netherlands, July 2005.

[5] J. Golbeck and J. Hendler. Inferring binary trust relationships in web-based social networks. *ACM Trans. on Internet Technology*, 6:497–529, 2006.

[6] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *Proc. of WWW*, pages 403–412, New York, USA, May 2004.

[7] M. A. Hasan, V. Chaoji, S. Salem, and M. Zaki. Link prediciton using supervised learning. In *Workshop on Link Analysis, Counter-terrorism and Security (SDM 2006 workshop)*, Bethesda,MD, April 2006.

[8] A. Josang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43:618–644, 2007.

[9] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputaion management in p2p networks. In *Proc. of WWW*, pages 640–651, Budapest, Hungary, May 2003.

[10] Y. A. Kim, M.-T. Le, H. W. Lauw, E.-P. Lim, H. Liu, and J. Srivastava. Building a web of trust without explicit trust ratings. In *Workshop on Data Engineering for Blogs, Social Medial and Web 2.0 (ICDE 2008 workshop)*, Mexico, June 2008.

[11] P. Nurmi. Perseus - a personalized reputation systems. In *Proc. IEEE/WIC/ACM Int'l Conference on Web Intelligence*, Nov. 2007.

[12] J. X. Parreira, D. Donato, C. Castillo, and G. Weikum. Computing trusted authority scores in peer-to-peer web search networks. In *Proc.of the 3rd int'l workshop on Adversarial information retrieval on the web*, pages 73–80, Banff, Canada, May 2007.

[13] D. Quercia, S. Hailes, and C. Licia. Lightweight distributed trust propagation. In *Proc. of IEEE ICDM*, Omaha, USA, October 2007.

[14] M. Richardson, R. Agrawak, and P. Domingos. Trust management for the semantic web. In *Proc. of 2nd Int'l Semantic Web Conference*, pages 351–368, Florida, USA, October 2003.

[15] S. Toivonen, G. Lenzini, and I. Uusitalo. Context-aware trust evaluation functions for dynamic reconfigurable systems. In *Proc. of WWW*, Edinburgh, UK, May 2006.

[16] Y. Wang and F.-R. Lin. Trust and risk evaluation of transactions with different amounts in peer-to-peer e-commerce environments. In *Proc. of IEEE Int'l Conference on e-Business Engineering*, pages 102–109, Shanghai, China, October 2006.