Content-based Trust Mechanism for E-commerce Systems

Yu Zhang¹, Huajun Chen¹, Xiaohong Jiang¹, Hao Sheng¹, Linhua Zhou¹, Tong Yu¹
¹ Zhejiang University, Hangzhou 310027, China
Email: yzh@zju.edu.cn

Abstract

Previous trust mechanisms are mainly focused on reputational models based on explicit trust ratings. However, the large amount of system-provided and user-generated content published on Web is often ignored. These information is very important for providing more accurate, fine-grained and efficient trust management. In this paper, we propose a content-based trust mechanism and introduce trust reasoning functionalities which can infer a user’s interests and expertise. We present a new method to calculate user similarity based on their feedbacks on reviews to reduce data sparseness of User Similarity matrix. We pinpoint several characteristics of trust which have been overlooked previously: trust is field-dependant, similarity-sensitive, personalized and etc. we differ various trust levels between pairs of users to support more accurate and efficient trust calculation. Although we choose Epinions as our targeted application, the mechanism proposed in this paper are also applicable to other e-commerce systems.

1 Introduction

Trust plays an important role in e-commerce applications. Which seller is more trustworthy to deal with? Whose review tells the truth? Is it safe to input the credit number and password into the website? To solve these problems, many researchers have proposed various trust mechanisms which mainly focused on reputational models based on explicit ratings [14, 13, 5, 6]. However, the large amount of system-provided and user-generated content published on Web is somewhat overlooked. These information are very essential to determine pairwise trust relationships and trust values. In this paper, we propose content-based trust mechanism to support more accurate, fine-grained and efficient trust management for e-commerce applications.

In this paper, we focus on Epinions [4], which is a successful product review website. On Epinions, users are allowed to specify whom to trust and build a personal web of trust. Web of trust is a network of reviewers whose reviews and ratings a user has consistently found to be valuable. According to web of trust, the system predicts how helpful a review will be to a user and promotes the reviews of trusted members, therefore he can find what he is looking for more easily and gets the most out of his time on the website.

Many researchers have proposed trust models based on web of trust of Epinions [10, 11, 9, 12], however, there are three major problems with these models:

1. First, since Epinions does not provide the functionality for a user to specify in which category he trusts another user, so all the above models assume that trust exists in all fields. This is simply not the truth. Alice trusts John’s point of view in computer hardware, however she can not appreciate his tastes of music. If we recommend John’s favorite songs to Alice, it is probably a waste of time and energy. From web of trust, we only know that Alice trusts John, but in which category she trusts him can not be obtained solely from the structure of the online social network.

2. Second, the search algorithm of these models is quite low efficient. Suppose that Alice trusts 150 users on Epinions, when she asks suggestions about piccolo trumpet, many previous models expand out from Alice and search all the paths of her 150 neighbors to get required information. The number of paths increases exponentially and the algorithm has a very high time complexity. In fact, only a few people in her trust network know about this kind of instrument. Searching all the paths of her neighbors leads to unnecessary high overhead of the whole system.

3. Third, there is no explicit information to delineate the variation of trust scales. For example, Alice has 150 friends in her “trusts” list, previous models treat each of her friends equally when assigning trust values. Statistically speaking, it is not possible that Alice trusts each of her 150 friends in web of trust to the same extent. Without differentiations over trust levels, the predicted trust values are usually not accurate.

As remarked earlier, the above implicit trust information can not be obtained solely from the structure of the online
trust network, therefore, we need to extract the content from the online system and reason about the trust relationships. In order to make up for such disadvantages, we proposed the content-based trust mechanism and trust reasoning functionalities for e-commerce applications.

2 Properties of Trust

In [2], the authors present the following properties such as: trust is subjective; trust is dynamic and non-monotonic; trust is based on prior experiences, etc. In this section, we will illustrate some unique features that are somewhat overlooked before.

- Trust is field-dependent. I trust you in solving mathematical problems, but I do not trust you in mending a car.
- Trust is similarity-sensitive. Consumers feel the most credible source for information about a company and products is a “person like themselves” [1]. Like-minded people tend to buy the same product.
- Trust is personalized. With various characters and life experience, different people have different preferences in choosing whom to trust. I may trust this group of users, you may trust another group of users. Even if we trust the same user, probably differ over trust levels.
- Trust is a community phenomenon. Trust is heavily linked to the development, fostering and maintenance of community relationships.

The term “trust” contains two meanings: first, whether a user tells the truth about the products they purchased or the services they used. The first meaning corresponds to honesty and credibility of a person or information source. Second, if one user thinks highly of a product, will I feel the same way as he was when using the same product? The second meaning relates to personal preferences and similarity between pairs of users. Both of the two aspects are important to trust management, and our proposed content-based trust mechanism try to meet the requirements of both aspects.

3 Relationships on Epinions

Generally speaking, there are only two concepts users and products on Epinions. Each concept can be divided into several categories. Among these entities, there are some kinds of relationships between them:

- “trusts”: When a user on Epinions consistently gives you good advice, you’re likely to trust that person’s recommendations in the future. You can add this person to your “trusts” list.
- “trusted-by”: When other users who have added you to their web of trust, you are trusted by them and appear on the list of users who trust you.
- “blocks”: When you encounter a user whose reviews are consistently invaluable, you can add that user to your “blocks” list. The “blocks” list makes it less likely that you will encounter recommendations you do not value in the future.
- “review”: You can describe your experience using this product or service – the pros and cons of the product or service.
- “comment”: You can make comments on a review. In fact, Epinions provides this platform for author and other readers to communicate with each other directly.
- “rate”: When you read reviews by other users, you can rate the reviews according to its value to others.

In the process of exploring the social media, users on Epinions are adding metadata in the form of:

- users’ reviews and ratings on products;
- users’ feedbacks and comments on reviews;
- social network of trustworthy friends;

Each type of the metadata allows users to leverage and share the knowledge and expertise of others. In this paper, we extract content from the website and mainly focus on “trust”, “review” and “rate” relationships between pairs of users.

4 Extract information from the content of Epinions

In this section, we will present how to extract the following trust related information from the content of Epinions (See Figure 1):

- Trust relationships
- Users’ interests and preferences
- User Similarity Matrix

4.1 Web of Trust

Web of trust can be obtained directly by crawling each user’s “trusts” and “trusted-by” lists from his home page on Epinions. Viewing users as vertices \( V \), trust relationship as directed edge \( E \), we obtain a directed graph \( G = (V, E) \).
4.2 Users’ interests and preferences

Users’ interests and preferences can be obtained by using explicit and implicit information from Epinions.com.

4.2.1 Explicit information

A small percentage of users play certain roles such as Category Lead, Top Reviewer and Advisor in specific categories on Epinions. Category Leads are active members who help Epinions oversee a particular category. Top Reviewers are active members who help shoppers find the best products on Epinions by writing high quality reviews in their category of expertise. Advisors are active members who help shoppers find the best content on Epinions by rating reviews in their category. They take different responsibilities and have greater influence than common users on local online community, so we call them super users. In this paper, we adopt role-based trust reasoning to infer a user’s interests and expertise in a specific category (we will illustrate role-based trust reasoning function in more details in Section 5.1). A user’s role on Epinions indicates his interests and expertise within a field. The strategy is that “if a reviewer is a super user in a specific category \( c_k \), then he must be interested in \( c_k \) and also an expert in this field”.

Another way to detect a user’s interests and expertise is from his own self-introduction. For example, on user “bryan_carey’s” homepage, he wrote that “I like to travel and I am an avid lover of beer. I estimate that I have tasted around 1000 to 1200 different beers.” From the content of the above sentences, we can tell that “travel” and “beers” are two of his most interested categories, and certainly he is a beer expert after tasting more than 1000 different beers. We can exploit a number of techniques based on data mining and natural language processing methods to mine a user’s interests and expertise [7].

4.2.2 Implicit information

Implicit information about a user’s interests and preferences can be obtained by analyzing the reviews and feedbacks contributed by users. When a user writes a review about a product or a service, he is either interested in that field or he once purchased the product or used the service. According to archive reviews, we can tell a user’s interested fields and his preferences in these fields. Under ordinary circumstances, users write a few (usually one or two) reviews about products they bought or services been served. For example, I write a review about my Nokia mobile phone. If a user writes hundreds of reviews on various of mobile phones, he must be obsessed with mobile phones and certainly has much more knowledge in this field than me. We suppose that the more reviews a user written in a specific category, the more likely he is interested in this field. We extract the title and category tag of each review, and then use simple statistical method to find out a user’s interested categories.

We can also infer a user’s interests by crawling his feedbacks on other users’ reviews from Epinions. Notice that many users read many reviews written by others, but write few by themselves. Therefore, it is hard to tell a user’s interested areas by their written reviews. However, we can obtain the information through a users’ feedback pattern. If a user often reads and offers feedbacks on reviews in a specific category, he must be interested in this area.

4.3 User Similarity Matrix

When two users share similar interests and preferences, they are more likely to influence each other. When a user writes a review or makes a recommendation, similar users possibly find it helpful and are more likely to purchase the product.

Traditional Collaborative Filtering algorithm for a recommendation system computes similarity value for pairs of users based on their ratings to products. When two users have rated more than two common products, user similarity can be calculated. However, the User Similarity matrix of Epinions is very sparse. Therefore, in this paper, we propose a new method to calculate user similarity and combine two kinds of user similarity to reduce data sparseness. When two users have rated two or more common products, then we can calculate user similarity in product ratings. When a user gives feedback on another reviewer’s reviews, then we can calculate user similarity in opinions.

Note that according to a user’s feedbacks on another reviewer’s reviews, user similarity in opinions is a direct measure of whether the user trusts in the reviewer’s point of
User similarity in product ratings can be calculated in a similar way. Epinions let users rate each product on a scale of 1 to 5 (1 indicates the worst and 5 indicates the best). In category $c_k$, if user $u_i$ and $u_j$ have rated $h$ products in common, then their ratings on these products can be regarded as a $h$-dimensional vector $\gamma$ and $\delta$. We also treat vectors as point coordinates in space and exploit similar method illustrated above to calculate user similarity in product ratings:

$$S_{c_k}^{u_i}(u_i, u_j) = 1 - \frac{\sqrt{\gamma_i - \delta_i)^2 + \cdots + (\gamma_h - \delta_h)^2 + (\gamma_l - \delta_l)^2}}{d_{max}}$$

(2)

where $S_{c_k}^{u_i}(u_i, u_j)$ represents user similarity in product ratings between user $u_i$ and $u_j$ in category $c_k$. $d_{max}$ denotes the maximum distance between two vectors (If $u_j$ rates each of the products in the set with a highest rating, while $u_i$ rates each of the products in the set with a lowest rating, then the distance is the maximum and vice versa).

Suppose there are $N$ users in the network, then we will have $N \times N$ User Similarity matrix $S_{c_k}$ in which the $j$th row denotes the similarity values of $j$th user against every other user in the network. If only one of the similarity relations exists between two users $u_j$ and $u_i$, then we assign the similarity value to item $S_{c_k}(j, i)$ in the matrix. If both of the two similarity relations exist, then we assign the maximum similarity value to item $S_{c_k}(j, i)$. If none of the relation exists, then the value of $S_{c_k}(j, i)$ is zero.

There are two benefits when making use of users’ feedbacks to calculate users’ similarity: first, many users take the initiative to read a user’s reviews and offer positive feedbacks, they are more likely to trust in his future reviews. Second, focusing on users who offer feedbacks is more easily to detect small communities of similar interests. Trust information is propagated among people within the community and quite important to the development of the community. Users who usually communicate with each other through reading reviews and offering feedbacks in a specific category are familiar with each other and naturally form a small community of similar interests.

## 5 Trust Reasoning Functionality

For online community of e-commerce systems, we can reason about trust information by analyzing the content from Epinions. In this section, we introduce the trust reasoning functionality of content-based trust mechanism.

First, we define the following notations: $u_i$ denotes a user on Epinions, $c_k$ denotes a specific category. $P(c_k)$ denotes the set of most popular authors in category $c_k$, $A(c_k)$ denotes the set of Advisors in category $c_k$. $T(c_k)$ denotes the set of Top Reviewers in category $c_k$, and $L(c_k)$ denotes the set of Category Lead in category $c_k$. We use $I(c_k)$ to illustrate the set of users who are interested in category $c_k$, and $E(c_k)$ to indicate the set of users who are experts in $c_k$. 

### Table 1. Review ratings of different levels

<table>
<thead>
<tr>
<th>Rating levels</th>
<th>Numerical Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not Helpful</td>
<td>-1</td>
</tr>
<tr>
<td>Not Rated</td>
<td>0</td>
</tr>
<tr>
<td>Somewhat Helpful</td>
<td>1</td>
</tr>
<tr>
<td>Helpful</td>
<td>2</td>
</tr>
<tr>
<td>Very Helpful</td>
<td>3</td>
</tr>
</tbody>
</table>

Note that usually a rater did not rate all of a reviewer’s reviews in a category. We assume that the rater is not interested in those unrated reviews, so we assign 0 to these reviews as the feedback rating.

According to statistics, we found that some users frequently read a key user’s reviews and always gave him positive feedback ratings to the content. If a user constantly regards another reviewer’s reviews as “very helpful”, which not only indicates that they are both interested in the same field, but also shows that they are like-minded people.

Let $l$ be the number of reviews written by user $u_i$ in a specific category $c_k$. Then user $u_j$’s ratings to $u_i$’s reviews can be regarded as a $l$-dimensional vector $\beta$. To evaluate the similarity between two users $u_i$ and $u_j$, we regard $u_i$ himself as a rater too. From $u_i$’s perspective, his own reviews always reflect his own personal preferences, therefore we assume that $u_i$ gives the highest rating “very helpful” to each of his own reviews. Thus, we use a constant vector $\alpha = [3,3,\ldots,3]$ to represent it.

To calculate user similarity, we regard vector $\alpha$ and $\beta$ as the point coordinates in space. Then we exploit Euclidean distance between two points to calculate dissimilarity, the user similarity in category $c_k$ can be calculated as follows:

$$S_{c_k}^{u_i}(u_i, u_j) = 1 - \frac{\sqrt{(\beta_1 - \alpha_1)^2 + \cdots + (\beta_k - \alpha_k)^2 + (\beta_l - \alpha_l)^2}}{d_{max}}$$

(1)

where, $S_{c_k}^{u_i}(u_i, u_j)$ denotes user similarity in opinions between user $u_i$ and $u_j$ in category $c_k$. $\alpha$ and $\beta$ denote the $k_{th}$ value of vector $\alpha$ and $\beta$ respectively. $d_{max}$ denotes the maximum distance between two vectors (If $u_j$ rates every of $u_i$’s reviews as “not helpful”, then the distance is the maximum).
We let \( u_j \xrightarrow{T} u_i \) denote \( u_j \) trusts in \( u_i \). The trust relationships can be obtained directly from web of trust on Epinions. We let \( u_j \xrightarrow{R(c_k)} u_i \) denote that \( u_j \) has read \( u_i \)'s reviews in category \( c_k \), while \( u_j \xrightarrow{F(c_k)} u_i \) denotes that \( u_j \) has never read \( u_i \)'s reviews in category \( c_k \). We use \( u_j \xrightarrow{F(c_k)} u_i \) to denote that \( u_j \) offers feedbacks on \( u_i \)'s reviews in category \( c_k \), while \( u_j \xrightarrow{R(c_k)} u_i \) denotes that \( u_j \) has never offered feedback on \( u_i \)'s reviews in category \( c_k \). We adopt two kinds of trust reasoning in our mechanism: one is role-based trust reasoning, the other is behavior-based reasoning. We will introduce them in the following subsections.

5.1 Role-based Trust Reasoning

Role-based trust reasoning uses author's role or membership in the online community to infer trust relationships. Since trust is field-dependent, we use the following clauses to determine a user’s interests:

1. \( u_i \in P(c_k) \Rightarrow u_i \in I(c_k) \)
2. \( u_i \in A(c_k) \Rightarrow u_i \in I(c_k) \)
3. \( u_i \in T(c_k) \Rightarrow u_i \in I(c_k) \)
4. \( u_i \in L(c_k) \Rightarrow u_i \in I(c_k) \)

The reasoning strategy of (1)~(4) is that if a user plays a certain role in a specific category \( c_k \) (such as Lead, Top Reviewer, Advisor, etc.), then he or she must be interested in that category \( c_k \).

We can also make use of a user’s role to infer his expertise in a specific category. Example rules are given as follows:

5. \( u_i \in P(c_k) \Rightarrow u_i \in E(c_k) \)
6. \( u_i \in A(c_k) \Rightarrow u_i \in E(c_k) \)
7. \( u_i \in T(c_k) \Rightarrow u_i \in E(c_k) \)
8. \( u_i \in L(c_k) \Rightarrow u_i \in E(c_k) \)

The reasoning strategy of (5)~(8) is that if a user plays an important role and take certain responsibilities such as advising other users about how to write reviews or in charge of the management of the whole category, then he or she must be an expert in that area.

We can also use role-based reasoning to determine trust relationships. Example rules includes: “Prefer to trust the reviews offered by reviewers rather than the advertisement by manufacturers” [3]. “Prefer to trust the reviews written by a top reviewer rather than a common user.”, etc.

5.2 Behavior-based Trust Reasoning

Behavior-based trust reasoning is to infer trust related information according to a user’s behavior in the online community. Example rules are as follows:

1. If \( u_i \) has never written reviews or offered feedbacks in category \( c_k \), then we regard that \( u_i \) is not interested in \( c_k \), and of course he or she is not specialized in this category either.

2. \( u_j \xrightarrow{T} u_i \land (u_j \xrightarrow{F(c_k)} u_i \lor u_j \xrightarrow{R(c_k)} u_i) \Rightarrow u_j \xrightarrow{T(c_k)} u_i \)

Consider that \( u_j \) is interested in \( c_k \) and \( u_j \) trusts in \( u_i \). However, \( u_j \) has never read nor given feedback on \( u_i \)'s reviews in this specific category. Generally speaking, we regard that a user added another user to his web of trust since he had found most of or at least some of the users’ reviews were helpful. If a user \( u_j \) has never paid attention to \( u_i \)'s reviews in this category, then \( u_j \) added \( u_i \) to his web of trust probably due to trusting in \( u_i \)'s reviews in some other categories. Therefore, \( u_i \) has little influence on \( u_j \) in category \( c_k \). We regard that \( u_j \) does not trust in \( u_i \) in this field.

3. \( u_j \xrightarrow{T} u_i \land (u_j \xrightarrow{F(c_k)} u_i \lor u_j \xrightarrow{R(c_k)} u_i) \Rightarrow u_j \xrightarrow{T(c_k)} u_i \)

The strategy of this clause is: if user \( u_j \) trusts \( u_i \) (\( u_j \) has added \( u_i \) in his “trusts” list), meanwhile \( u_j \) reads \( u_i \)'s reviews in category \( c_k \) and offers feedbacks on these reviews, then \( u_j \) trusts \( u_i \) in category \( c_k \). We exploit this reasoning rule to infer from general trust relationship to category-specific trust relationship which is very important for trust management.

6 Construct the Combined Network

In this section, we combine web of trust with User Similarity matrix to construct a category-specific trust network. According to trust reasoning functionality introduced in Section 5, if \( u_j \) is not interested in a specific category \( c_k \), then the directed edge \((u_j, u_i)\) (\( u_j \) trusts in \( u_i \)) in web of trust is removed from the combined network.

In category \( c_k \), if \( u_j \) has given feedbacks on \( u_i \)'s reviews or they have rated the same set of products, but \( u_i \) was not in \( u_j \)'s “trusts” list, then we add a directed edge \((u_j, u_i)\) to the combined network and assign the user similarity value to \((u_j, u_i)\) as the edge weight. This kind of users are often ignored by previous trust models since they only focused on the explicit network – web of trust. Unlike other trust models assigning the same trust value to every neighbor of the source user, in this paper we take similarity between pairs of users as the criterion for one users’ influence on another based on the study in sociology [8]. The more similar of two users, the more likely they may influence each other in accepting recommendations or making purchase decisions. If user similarity in product ratings and user similarity in opinions coexist, then we adopt the maximum value of the two as the edge weight.

If user \( u_j \) trusts in \( u_i \) (\( u_i \) is in \( u_j \)'s “trusts” list), and meanwhile the item \( S_{c_k}(j, i) \) in User Similarity matrix is non-zero (\( u_i \) and \( u_j \) has similarity relations), then according to rule (3) of behavior-based trust reasoning, we can infer that \( u_j \) trusts in \( u_i \) in category \( c_k \), that is \( u_j \xrightarrow{T(c_k)} u_i \). Since both the “trust” and “similar” relations exist together,
we augment the corresponding item in User Similarity matrix and assign the value $\min(w_r \times \alpha, 1) (\alpha$ is the augment parameter and $\alpha > 1$) to edge $(u_j, u_i)$ be the weight of the combined network. If we can determine that $u_j$ trusts in $u_i$ in category $c_k$, then $u_j$'s recommendations and suggestions in $c_k$ are much more reliable to $u_j$. After constructing the combined network, we can take advantage of various trust models for trust calculation and rating predictions.

7 Conclusion and Future Work

In this paper, we present content-based trust mechanism for e-commerce systems. Our contribution is multi-fold: first, we propose to exploit user-generated and system-provided content from e-commerce website for more accurate and fine-grained trust management. Secondly, we pinpoint several characteristics of trust which have been overlooked previously: category-dependant, similarity-sensitive, personalized and etc. Third, we introduce role-based and behavior-based trust reasoning to infer a user’s interests and expertise. With these information, we can infer from the general trust relationship to category-specific trust relationship, which can provide more accurate trust prediction. Forth, we take advantage of feedbacks on reviews to calculate user similarity to reduce data sparseness. Fifth, we differ various trust levels between pairs of users to support more efficient trust calculation. Although we choose a product review website Epinions as our example, the methods and approaches can be applied to other major e-commerce applications as well.

An interesting direction for future work is opinion mining and sentiment analysis of user-generated content to have a more detailed evaluation of one user’s trust on another. Then we can set up a more complete trust model for online users. Since there are huge amount of high-quality content available on Web, it provides us a very good platform to conduct further research.

8 Acknowledgments

This research was funded in part by the following projects under grant number: NO.2006BAH02A01, NO.51306030101, NO.2006AA01A123, NO. NCET-04-0545 and NO.NSFC60533040.

References


