Zhang Y, Yu T. Mining trust relationships from online social networks. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 27(3): 492–505 May 2012. DOI 10.1007/s11390-012-1238-8

Mining Trust Relationships from Online Social Networks

Yu Zhang[1,*] (张 宇), *Member*, *CCF*, *ACM*, and Tong Yu[2] (于 彤)

[1] *School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China*

[2] *College of Computer Science, Zhejiang University, Hangzhou 310027, China*

E-mail: yzh@zstu.edu.cn; ytcs@zju.edu.cn

Received August 31, 2011; revised December 18, 2011.

Abstract With the growing popularity of online social network, trust plays a more and more important role in connecting people to each other. We rely on our personal trust to accept recommendations, to make purchase decisions and to select transaction partners in the online community. Therefore, how to obtain trust relationships through mining online social networks becomes an important research topic. There are several shortcomings of existing trust mining methods. First, trust is category-dependent. However, most of the methods overlook the category attribute of trust relationships, which leads to low accuracy in trust calculation. Second, since the data in online social networks cannot be understood and processed by machines directly, traditional mining methods require much human effort and are not easily applied to other applications. To solve the above problems, we propose a semantic-based trust reasoning mechanism to mine trust relationships from online social networks automatically. We emphasize the category attribute of pairwise relationships and utilize Semantic Web technologies to build a domain ontology for data communication and knowledge sharing. We exploit role-based and behavior-based reasoning functions to infer implicit trust relationships and category-specific trust relationships. We make use of path expressions to extend reasoning rules so that the mining process can be done directly without much human effort. We perform experiments on real-life data extracted from Epinions. The experimental results verify the effectiveness and wide application use of our proposed method.

Keywords trust mining, trust reasoning, implicit trust, category-specific trust, semantics

1 Introduction

With the rapid development of information technology and widespread use of Internet, applications such as email, e-commerce, online payment and instant messaging have become an indispensable part of people's daily life. People are connected to each other through a variety of mutual relationships, forming many large, complicated and content-rich online social networks. As it gets easier to add information to the web via html pages, wikis, and blogs, it gets tougher to distinguish accurate information from inaccurate or untrustworthy information. Trust is essential to secure and high quality interactions in online social networks.

Existing trust research for online community is mainly based on explicit reputation information. However, only utilizing reputation information is far from sufficient. A large number of implicit trust relationships are more or less ignored. Meanwhile, due to lack of category information, many existing trust models often regard that trust exists in all categories universally, which causes trust relationships being over-generalized. Using generalized trust relationships for trust calculation, often results in low accuracy and high time complexity.

To solve the above problems, we propose a semantic-based reasoning mechanism to mine trust relationships from online social networks automatically. This mechanism utilizes Semantic Web technologies to define a domain ontology, then makes use of OWL (Web Ontology Language)/RDF (Resource Description Framework) language for knowledge representation of web data, and extracts trust-related information from the data. Based on the above information, we define OWL-based trust reasoning rules. Using these rules, we manage to reason about users' interested categories, discover implicit trust relationships, and infer from generalized trust relationships to *category-specific* trust relationships, thus supporting more accurate and fine-grained trust calculation for online social networks.

Regular Paper

This work is supported by the National Natural Science Foundation of China under Grant No. 61100183, the Natural Science Foundation of Zhejiang Province of China under Grant No. Y1110477, and the Science Foundation of Zhejiang Sci-Tech University under Grant No. 0907838-Y.

*Corresponding Author

2 Related Work

Earlier work of mining trust from online social networks mainly relies on an existing web of trust that has explicit trust from one user to another. Guha *et al.* propose a trust propagation model to determine trust relationships between two users through such a web of trust[1]. Other work on propagating trust through web of trust include [2-4]. Since the above approaches only rely on the structure of trust network, the accuracy cannot be guaranteed due to lack of context and content data such as category or topic information, users' roles, users' behaviors and user similarity. Nowadays, incorporating context or content data for mining social networks and social web has been widely used in many research directions and proven to be effective in these fields, such as trust management[5], social influence analysis[6-7], rating prediction[8], recommender system[9], and targeted advertising[10].

Current mainstream solutions for online social relationships mining are statistical learning-based approaches. For example, Tang *et al.* incorporate social theories into a semi-supervised learning framework, which effectively improves the accuracy of inferring various types of social ties over large social networks[11-12]. Leskovec *et al.* construct predictors using theories of signed social networks and node degree to predict positive and negative links in online social networks[13-14]. Different from the above approaches, we exploit the Semantic Web technologies in this article to describe the problem and incorporate human knowledge into reasoning rules to infer trust relationships automatically.

Zolfaghar and Aghaie propose an ensemble system consisting of classification models such as SVM (support vector machine), RBF (radial basis function) Network, decision tree and logistic regression to predict trust and distrust relations among Epinions online users[15]. One shortcoming of their approach is that they only focus on user ratings and do not pay any attention to the category information of user relationships.

Lim *et al.* present a trust antecedent framework which derives ability, benevolence and integrity as the three key factors in trust formation[16-17]. Several trust ranking models are proposed using above features. One problem of their approach is that it is difficult to acquire all the three key factors in real online communities.

Nguyen *et al.* develop measures for predicting if a trustee will return trust to his/her trustor given that the latter initiates a trust link earlier[18]. Different from Nguyen's work which has known unilateral trust relationship beforehand, we also manage to mine relationships when there are no explicit relationships between users at all.

Liu *et al.* propose a supervised learning method that automatically predicts trust relationships between a pair of users using evidence from actions of individual users and interactions between users[19-20]. They develop a taxonomy to acquire relevant features from user attributes and user interactions, which is similar to our defined domain ontology in Section 4. One key difference is that our method exploits the Semantic Web technology to support automatic trust reasoning, which has more versatility in different kinds of online applications. Another key difference is that they mainly focus on users' behaviors, while we also pay attention to users' roles and user-generated content.

Matsuo and Yamamoto take advantage of features extracted from user profiles, product reviews and existing trust relations to predict trust between users[21]. The general idea is similar to our proposed mechanism in this paper. However, the key difference between their method and ours is that we incorporate category information with trust and support automatic trust reasoning based on predefined ontology.

Skopik *et al.* propose a system which determines trust relationships between users by mining communication data[22]. They analyze the structure of discussions, examine interaction patterns and infer users' social roles. Their approach mainly focuses on discussions among users, but a large amount of other system-provided and user-generated content in online social networks are more or less overlooked.

Compared with earlier work, we have the following innovations. First, we pay attention to a variety of context and content data, such as users' roles, users' rating, users' behaviors and category information. We comprehensively utilize different dimensions of information for mining trust relationships. Second, we exploit the Semantic Web technology to build domain ontology for knowledge expression and description of trust relationships. Third, we propose a semantic-based trust reasoning mechanism to automatically infer users' personal interests, and mine the implicit trust relationships and category-specific trust relationships. Fourth, we make use of path expressions to extend trust reasoning rules, which makes complex queries can be processed directly using derived knowledge in knowledge base. Therefore, we can carry on automatic trust mining according to predefined rules without much human effort.

3 Overview of Epinions

In this paper, we choose Epinions as the target application, which is a successful product review website[23]. Epinions helps customers make informed buying decisions through providing unbiased advice, in-depth product evaluations and personalized recommendations. Users are also 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, so that the user can find what he/she is looking for more easily and gets the most out of his/her time on the website.

Fig.1(a) shows a snapshot of a user's homepage on Epinions, which includes his/her web of trust, roles, activity summary and written reviews. From the reviews, we can extract product names, belonging categories, product ratings, overall review ratings, etc. Fig.1(b) shows detailed information of user-generated content. From top-left counter clock wise, part (A) presents users' ratings on a specific review. Different people may have different opinions on the same review, including

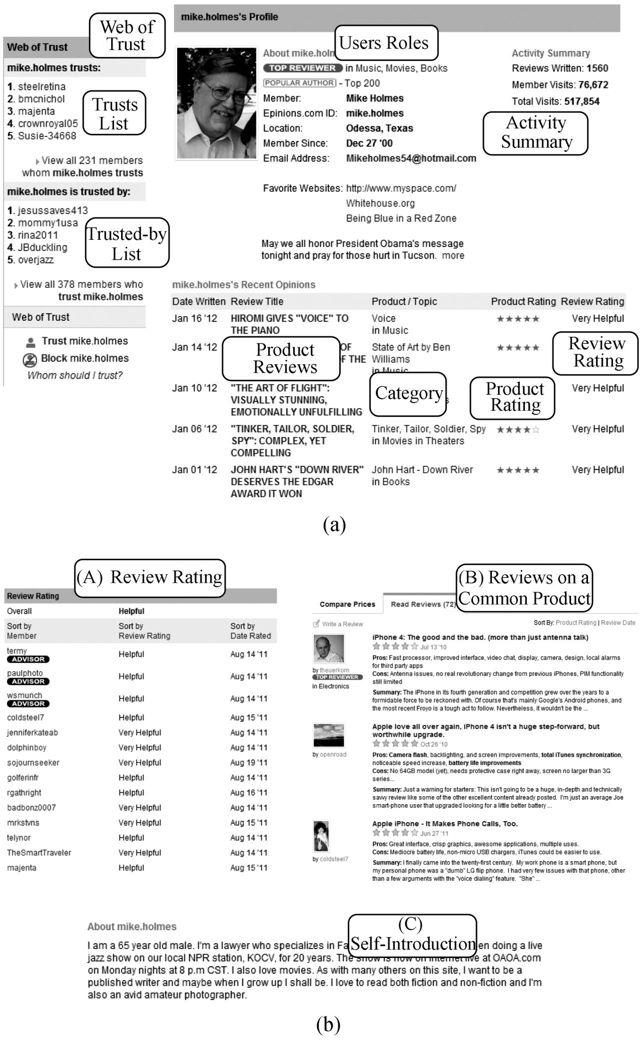

Fig.1. (a) Snapshot of an Epinions user's homepage. (b) User-generated content on Epinions.

"not helpful", "somewhat helpful", "helpful" and "very helpful". Part (B) presents several reviews on the same product from different users. Part (C) shows users' self-introduction. We can take good advantage of the above information to mine trust relationships from online social networks.

On Epinions, there are three main concepts: *users*, *products* and *reviews*. Among these entities, there are several kinds of relationships between them.

1) *Trusts*. When a user on Epinions consistently gives you good advice, you are likely to trust that person's recommendations in the future. You can add this person to your *trusts* list.

2) *Trusted-by*. When other users who have added you to their web of trust, you are trusted by them and appear on their *trusted-by* lists.

3) *Blocks*. When you encounter a user whose reviews are consistently invaluable, you can add that user to your *blocklist*. The *blocklist* makes it less likely that you will encounter recommendations you do not value in the future.

4) *Reviews*. You can describe the experience using a product or service through writing reviews: the pros and cons of the product or service.

5) *Rates*. When you read a review written by another user, you can rate the review according to its value to others.

6) *Comments*. You can make comments on a review. Epinions provides this platform for the author and other readers to communicate with each other directly.

In the process of mutual interactions, users on Epinions add metadata to the website in the form of: 1) reviews and ratings on products; 2) feedbacks and comments on reviews; 3) social network of trustworthy friends. Each type of metadata allows users to leverage and share the knowledge and expertise of others.

4 Domain Ontology for Online Social Networks

Ontology is defined as formal specifications of vocabularies that can be shared and reused among various communities and organizations. By using ontology, the web content and community context are machine understandable, which enables automatic detection of trust related information and trust relations. Ontology also provides a coherent categorization scheme for logical trust reasoning.

4.1 Domain Ontology Overview

In this paper, we make use of existing ontologies including FOAF (the Friend of a Friend Project)[24], SIOC (Semantically-Interlinked Online Communities

Project)[25] and GoodRelations (the Web vocabulary for e-commerce)[26] to build domain ontology, which provides main concepts and properties for online social networks. FOAF is used to describe people, such as personal information, the links between them and the things they create and do. SIOC is used to describe the structure and content of online communities, and to find related information and new connections between content items and other community objects. GoodRelations defines web vocabularies for e-commerce, which mainly describes products and services. We extend the above three existing ontologies to enrich the information described and make our domain ontology more universal for other online applications.

We utilize FOAF to describe users in the online community, exploit SIOC to depict various relations between content items and community objects, and make use of GoodRelations to describe products and services offered online. As shown in Fig.2, we extend the above ontologies by integrating trust relations with category information and build the domain ontology using Protégé.

In Fig.2, the rounded rectangles represent classes and the directed edge with an arrowhead indicates the relation between two classes. We will introduce the main concepts and relations in more details below.

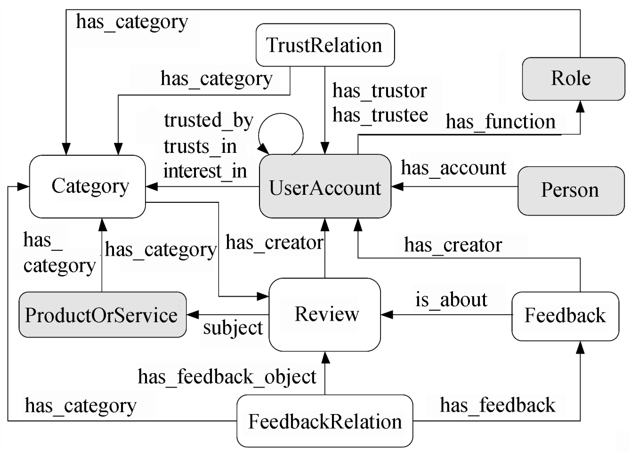

Fig.2. Domain ontology for online social networks.

4.2 Main Concepts

For the domain ontology, we can reuse the vocabularies defined in existing ontologies. For example, the domain ontology reuses the *Person* class from FOAF, the *UserAccount*, *Role*, *Category* classes from SIOC, and the *ProductorService* class from GoodRelations. We utilize "imports" statement to connect existing ontologies to our defined domain ontology. OWL allows to import the entire content from other ontologies using "imports" statement[27]. The defined ontology also inherits corresponding attributes and relations of the above classes from existing ontologies. We can define some new concepts to enrich the domain ontology as well.

4.2.1 *Users' Roles*

On Epinions, a user may play different roles, such as reviewer, rater and administrator. We define *reviewer* as the user who contributes reviews to Epinions and define *rater* as the user who offers feedback ratings to others' reviews. We also define *administrator* as the user who is in charge of the online community. The OWL definitions of the above users' roles are as follows:

```
ex:reviewer rdfs:subClassOf role
ex:rater rdfs:subClassOf role
ex:administrator rdfs:subClassOf role.
```

Each type of role has subclasses. For example, *reviewer* is composed of *top_reviewer*, *popular_author* and *common_reviewer*. *Rater* is divided into *advisor* and *common_rater*. *Administrator* is composed of *category_lead* and *system_administrator*. We take *top_reviewer* as an example and illustrate its OWL definitions below:

```
ex:top_reviewer rdfs:subClassOf ex:reviewer.
```

We exploit Boolean operator `AND` to integrate category information into users' roles. Therefore, a top reviewer in movie field can be defined as follows:

```
ex:top_reviewer_in_movie = ex:top_reviewer
&(ex:has_category value ex:Movie).
```

4.2.2 *Entity Classes*

For Epinions community, we define two entity classes: one is *Review*, the other is *Feedback*. *Review* refers to the articles written by a *Reviewer* about a *ProductorService*, denoted as follows:

```
ex:Review ex:has_creator value ex:UserAccount
ex:Review ex:has_category value ex:Category
ex:Review ex:subject value ex:ProductorService.
```

The entity class *Feedback* refers to the review rating offered by a *Rater* towards a user's *Review*. The OWL definition is as follows:

```
ex:Feedback ex:has_creator value ex:UserAccount
ex:Review ex:has_feedback value ex:Feedback.
```

4.3 Relations

We inherit corresponding relations between classes from existing ontologies and define new relations according to the application scenario. In this article, we define two kinds of relations: one is about users' interests, the other is about users' trust relationships. For reasoning purpose, we incorporate *category* information with the above relations.

4.3.1 Support N-ary Relations

The Semantic Web languages, such as RDF and OWL, can only describe binary relations (properties) between individuals. For example, an individual A and an individual B have a property P or an individual A has a property P with the value B (see Fig.3(a)). We can use this method to describe a trust relationship like "Bob trusts Peter", where Bob and Peter are two individuals and "trusts" is the relation between them.

In this paper, we want to incorporate category information into the trust relationship: Bob trusts Peter in movie field (see Fig.3(a)). That is, property P now is a relation among A, B and C. We exploit the method introduced in [28] and regard one of the individuals in the relation as the originator. For the above example, Bob is the originator. We create an instance that includes the originator object itself, as well as the additional information about the object. Individual Bob has a property *trusts* and *trusts* has another object *tr_1* which is an instance of the relation *Trust_Relation* (see Fig.3(b)).

Here we restrict the field of trust relations to be one value in class *Category* on Epinions. We also require that each *Trust_Relation* has exactly one value for *Category*. Note that the vocabularies in *Category* are predefined in domain ontology, which makes it possible for machines to understand and process.

4.3.2 Define Category-Specific Relations

Using the methods illustrated above, we can define *category-specific* relations between users. In this paper, we mainly focus on three types of relations: the first one is *interests_in* relation, the second one is *trusts_in* relation, and the third one is *feedbacks_on* relation.

Taking *movie* field for example, we define *user_interests_in_movie* to denote the user who is interested in the movie field.

```
ex:user_interests_in_movie = UserAccount
& (ex:interests_in value ex:Movie).
```

By analogy, we can define relation *trusts_in_movie* that indicates one user trusts another user in movie field, denoted as:

```
ex:trusts_in_movie = isa trusts_in
domain: ex:UserAccount
range: ex:UserAccount value: ex:Movie.
```

We can also define *feedbacks_on_movie_review* that denotes a rater gives feedback rating to a review in movie field. The OWL representation is as follows:

```
ex:feedbacks_on_movie_review =isa
ex:feedbacks_on
domain: sioc:UserAccount
range: movie_review.
```

Using Semantic Web technology, we manage to integrate category information into relations of the online social networks, which provides the basis for automatic semantic-based trust reasoning.

5 Semantic-Based Trust Reasoning

In this section, we propose an innovative semantic-based trust reasoning mechanism. Using this mechanism, we manage to reason about users' interests and expertise, identify implicit trust relationships and infer from over-generalized trust relationships to category-specific trust relationships. We will introduce more details below.

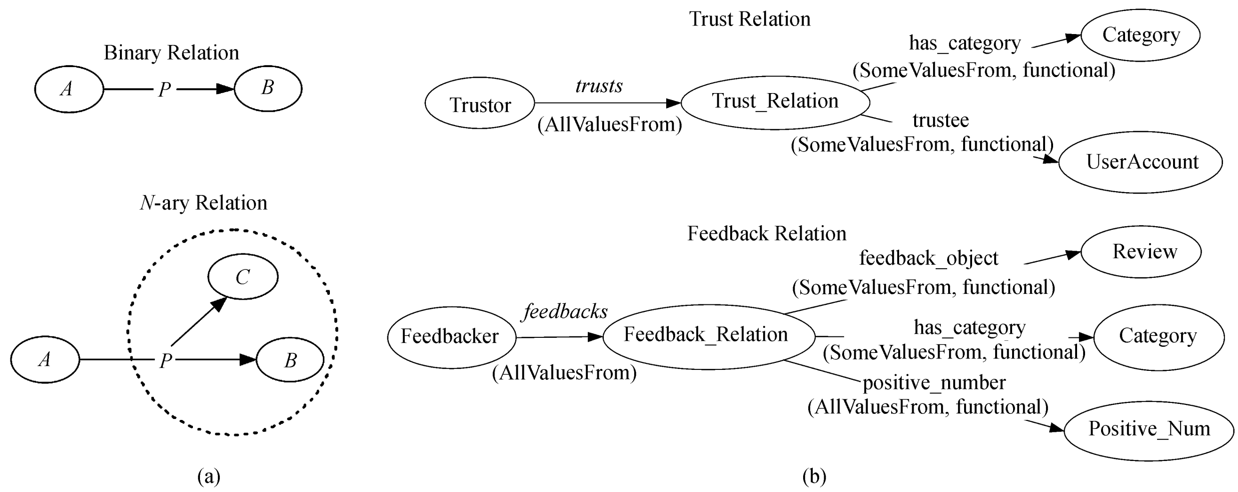

Fig.3. (a) Support n-ary relations. (b) Trust relation and feedback relation.

5.1 Reason About User's Interests

Personal interest is a key point that keeps online users to form a connected group in the online social networks. Users who share common interests are more inclined to form trust relationships. In this subsection, we illustrate how to infer a user's interests. We adopt two kinds of reasoning patterns: one is role-based reasoning, and the other is behavior-based reasoning.

First, we define the following notations. U denotes the set of user accounts on Epinions and u_i denotes a user ($u_i \in U$). c_k denotes a specific category. $T(c_k)$ denotes the set of Top Reviewers in category c_k. $A(c_k)$ denotes the set of Advisors in category c_k. $L(c_k)$ denotes the set of Category Leads in category c_k. $P(c_k)$ denotes the set of Popular Authors in category c_k. $I(c_k)$ denotes the set of users who are interested in category c_k.

5.1.1 Role-Based Reasoning About Users' Interests

Role-based reasoning uses users' role or membership in the online social networks to infer a user's interest. As known to all, a small percentage of users play certain roles such as Top Reviewer, Advisor, Category Lead and Popular Author in specific categories on Epinions. Top Reviewers are active members who help shoppers find the best products on Epinions by writing high quality reviews in their expert fields. Advisors are active members who help shoppers find the best content on Epinions by rating reviews. Category Leads are active members who help Epinions oversee a particular category. Popular Authors in specific categories are selected based on the members' total number of visits by other users in that category. They play different roles in the online community but they all have greater influence than common users, so we call them *super users*. We exploit the following reasoning rules to infer a user's interests:

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

The strategy of the above rules is that if a user plays a certain role in a specific category c_k (such as Top Reviewer, Advisor and Category Lead), then he/she must be interested in that category.

As remarked earlier, each kind of role is associated to class *Category* with the property *has_category*. For example, Peter is a Top Reviewer in *Movie* category, denoted as:

```
Individual(a:Peter
  type(a:top_reviewer_in_movie)).
```

According to the predefined ontology illustrated in Section 4, it is easy to come to the conclusion that Peter is interested in movie field automatically (see Fig.4), denoted as:

```
Individual (a:Peter type(
  a:user_interested_in_movie)).
```

5.1.2 Behavior-Based Reasoning About Users' Interests

Behavior-based reasoning is to infer information and trust relationships according to a user's behaviors in the online community.

A simple method to identify a user's interest is to analyze his/her browsing history. Due to privacy reasons, we are not able to obtain such data from Epinions. However, we can infer a user's interest according to his/her written reviews or his/her feedback ratings on others' reviews. The above data is visible to everyone and can be crawled directly from Epinions.

The general idea of behavior-based reasoning is straightforward. For example, if a user writes hundreds of reviews on various types of mobile phones, he/she

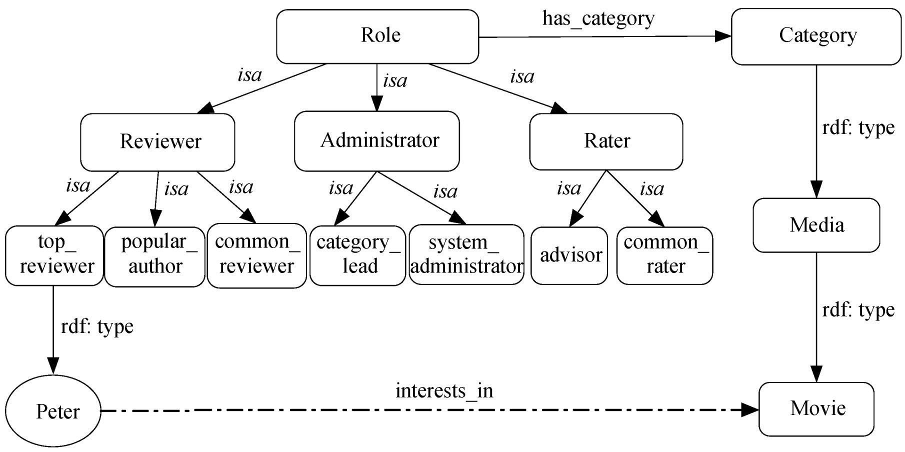

Fig.4. Role-based reasoning about a user's interest.

must be obsessed with mobile phones and certainly has much more knowledge in this field. We suppose that the more reviews a user wrote in a specific category, the more likely he/she is interested in that field. In this way, we are able to infer a user's interest according to his/her written reviews.

As known to all, most users on product review web sites often read many reviews written by others, but seldom write any by themselves. Therefore, for most online users, it is hard to identify their interests according to their written reviews. Under such circumstances, we can infer a user's interest according to his/her feedback ratings on others' reviews. If a user often offers feedback ratings on reviews in a specific category, then he/she is more likely to be interested in that field.

Below, we define two reasoning rules according to users' behaviors. The first rule is based on users' review writing behaviors, the second one is based on users' feedback behaviors.

Rule 1: if the number of reviews written by a user u_i in a specific category c_k exceeds the threshold h, then we regard that u_i is interested in category c_k. The value of parameter h can be adjusted accordingly.

Rule 2: if the number of feedback ratings given by a user u_i in a specific category c_k exceeds the threshold g, then we regard that u_i is interested in category c_k. The value of parameter g can be adjusted accordingly.

Below, we will give an example about how to infer a user's interests based on his/her online behaviors. Suppose that we want to know about Bob's personal interests. We can utilize reasoning Rule 1 illustrated above, which can be easily implemented using SPARQL query language (SPARQL Protocol and RDF Query Language) as follows:

```
SELECT ?category,COUNT(distinct *) as count
WHERE {
      ?review dc:creator ex:Bob.
      ?review ex:reviews ?product.
      ?product rdf:type ?category.
} GROUP BY ?category
HAVING (count(distinct *) > h ).
```

After query, the result turns out to be:

?category	?count
ex:Movie	6
ex:Music	7

According to the above query result, the system can automatically come to the conclusion:

```
ex:Bob ex:interests ex:Movie, ex:Music.
```

That is, Bob is interested in *Movie* and *Music*. Users' personal interests are closely related to trust relationships, small groups, word of mouth, and targeted marketing, etc. Therefore, we can save the query result into a new table named *Interests* in the knowledge base (KB). Knowledge base stores category-related theory, fact data, expert experience, and heuristic knowledge, which includes related definitions, theorems, algorithms and other common sense knowledge. When we need to search for users' personal interests in the future, we can directly refer to table *Interests*, which can greatly improve the query performance.

5.2 Infer Category-Specific Trust Relationships

On Epinions, we can obtain explicit trust relationships from web of trust. Due to lack of category information, we only know that user A trusts user B, however, we do not know in which categories that A trusts B. In this paper, we exploit semantic-based trust reasoning to infer category-specific trust relationships. The advantage of determining category-specific trust relationships is two fold. First, it can reduce the complexity of trust computation for a specific category. Second, it can provide online users with more valuable and pointed suggestions for decision-making.

Here, we define a reasoning rule to infer category-specific trust relationship between users. For $\forall u_i \in U$ and $\forall u_j \in U$, given a certain category c_k, if the following conditions can be satisfied simultaneously, 1) u_j has a generalized trust relations towards u_i (that is, u_i is in u_j'*s trusts list*), 2) u_i is interested in category c_k, 3) u_i is a *super user* in category c_k (such as Top Reviewer, Category Lead, Advisor, or Popular Author); then, we can infer that u_j trusts u_i in category c_k.

This reasoning rule can be implemented using the following SPARQL query:

```
CONSTRUCT {?trustor ex:trusts ?trustee}
WHERE {
?trustor ex:interestedInCategory ?category
?trustee ex:topReviewerInCategory ?category
}.
```

Below, we will illustrate an example on how to infer category-specific trust relationship. The known facts are: 1) Peter is a top reviewer in movie field and Bob is interested in movie field, denoted as *knowledge meta-data* X: *?u_i ex:top_reviewer_in ex:Movie. ?u_j ex:interested_in ex:Movie.* (see Fig.5(a)); 2) Peter is in Bob's *trusts* list, denoted as *knowledge meta-data* Y: *?u_j ex:trusts ?u_i* (see Fig.5(b)). The RDF representation of the above knowledge are as follows:

```
Individual(ex:Peter
  type(ex:top_reviewer_in_movie))
Individual(ex:Bob type(ex:common_user)
  value(ex:trusts ex:Peter))
ex:Bob ex:interests_in ex:Movie.
```

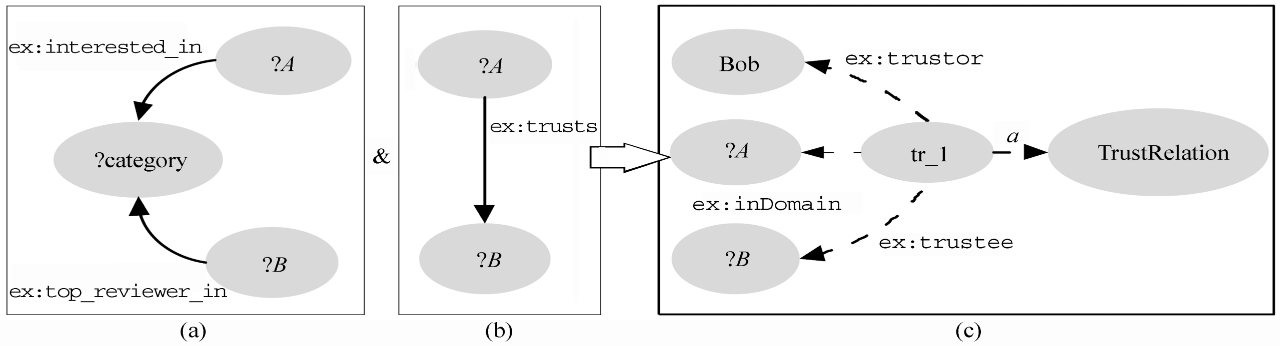

Fig.5. Reasoning pattern for category-specific trust relationships.

Using reasoning pattern $X + Y \Rightarrow Z$, we combine the two parts of *knowledge meta-data* and obtain new semantic knowledge Z (see Fig.5(c)), denoted as:

```
tr_1 a ex:TrustRelation;
ex:trustor ?A;
ex:trustee ?B;
ex:inDomain ?domain.
```

The dotted lines in Fig.5(c) indicate the derived knowledge obtained from semantic-based reasoning. According to the reasoning rule in knowledge base, we can come to the conclusion that Bob trusts Peter in movie field, denoted as:

```
Individual(ex:Bob type(ex:common_user)
  tr_1 a ex:TrustRelation
  trustor(tr_1,ex:Bob)
  trustee(tr_1,ex:Peter)
  trust_field((tr_1, ex:Movie).
```

5.3 Infer Implicit Trust Relationships

On Epinions, we observed that some of the users have offered many positive ratings towards a reviewer's reviews in a certain specific category, however they do not add that reviewer to their *trusts* list. Their online behaviors clearly show that they trust that reviewer's point of view in this field. If we only focus on those explicit trust relationships obtained directly from web of trust, the above important but implicit trust relationships are more or less overlooked. These trust relationships are important to accurate and fine-grained trust management.

In our daily life, if two persons reach agreement on a series of problems in a field, they are more likely to trust each other. Positive feedback rating is a special kind of evidence to show that the rater agree with a reviewer's point of view. Epinions allows users to offer four grades of ratings towards a review: "Very Helpful" (VH), "Helpful" (H), "Somewhat Helpful" (SH), and "Not Helpful" (NH). We add another rating grade "Neutral" (N) to illustrate that a user is either not interested in the review written or has no comment on it. Then we define a set γ to include all the feedback rating grades:

$$\gamma = \{\text{VH, H, SH, N, NH}\}.$$

In domain ontology, we exploit multiple non-intersected sub-classes to describe the above rating grades. The RDF representation is as follows:

```
Class(Review_Rating_Value
  equivalentClass unionOf
  (Very_Helpful Helpful
  Somewhat_Helpful Neutral Not_Helpful))
Class(Very_Helpful partial
  Review_Rating_Value)
Class(Helpful partial
  Review_Rating_Value)
Class(Somewhat_Helpful partial
  Review_Rating_Value)
Class(Neutral partial
  Review_Rating_Value)
Class(Not_Helpful partial
  Review_Rating_Value)
Class(Positive_Rating  range(
  unionOf(Very_Helpful Helpful))
  subClassOf (Review_Rating_Value)).
```

In this paper, we regard the union of "Very Helpful" (VH) and "Helpful" (H) as the set of positive feedback ratings (denoted as F). Therefore, in category c_k, the number of positive feedback ratings given by u_j to u_i can be calculated as follows:

$$Num^{c_k}_{u_j \to u_i}(\text{F}) = Num^{c_k}_{u_j \to u_i}(\text{VH}) + Num^{c_k}_{u_j \to u_i}(\text{H}).$$

Here, we define another reasoning rule according to users' feedback behaviors. If the number of positive feedback ratings given by user u_j to u_i in category c_k exceeds the threshold θ, then we regard that user u_j trusts u_i in category c_k. The value of parameter θ can be adjusted accordingly.

This reasoning rule can be easily implemented using SPARQL query language as follows:

```
CONSTRUCT{

   _tr1    a  ex:TrustRelation;
           ex:trustor ?feedbacker;
           ex:trustee ?reviewer;
           ex:inDomain ex:Movie.

} WHERE {
   ?feedback a ex:Feedback;
           ex:inDomain ex:Movie;
           ex:hasReviewer ?reviewer;
           ex:hasFeedbacker ?feedbacker;
           ex:rate ex:positiveRating.
} GROUP BY ?reviewer,?feedbacker
HAVING  (count(distinct *)> θ )
```

6 Extension of Reasoning Rules in Knowledge Base

In this paper, some reasoning rules can be implemented using SPARQL query directly, while others may require derived knowledge from knowledge base. For example, there are several records stored in knowledge base as follows:

```
ex:Bob ex:rates ex:Roman_Holiday.
ex:Roman_Holiday a ex:Comedies.
ex:Comedies isa ex:Videos&DVDs.
ex:Videos&DVDs isa ex:Movie.
ex:Movie isa ex:Media.
ex:Media isa ex:Category.
```

From the above data, machines only know that: "Roman Holiday is a comedies". However, machines do not know "Roman Holiday is a movie", which is quite obvious to us humans according to our common sense. Suppose that we want to know the number of positive review ratings given by Bob in movie field. It is difficult to obtain the results based on the above data records only using common SQL queries. (We need to perform multiple join operations among several tables in the underlying database, which leads to high time complexity and requires great human effort.) To solve this problem, we exploit path expressions to extend the reasoning rules in knowledge base. We will illustrate in more details below.

6.1 Path Expressions

Path expressions identify an object by describing how to navigate to it in a graph of objects. We take the previous query for example: how many review ratings are offered by Bob in movie field. We can rewrite the original query into a complex one with path expressions, which matches the query pattern in knowledge base directly (see Fig.6). The path composed by red solid lines can be transformed into a direct path using combination operator $a(isa)^{2*}$. While the path composed by green solid lines can be transformed into a direct path using another combination operator $(isa)^{2*}$. The combination operators is described as follows:

$$a(isa)^{m*} \circ (isa)^{n*} = (a \diamond \underbrace{isa \diamond \cdots \diamond isa}_{m}) \circ (\underbrace{isa \diamond \cdots \diamond isa}_{n}),$$

where, symbol $\circ$ denotes *jump operator*, which means to jump from one knowledge unit to another. A knowledge unit is represented as an RDF graph that is a subgraph in the knowledge base. Symbol $\diamond$ denotes *path connection operator*, which means to go from one entity to another. The jump operator is different from the path connection operator: jump operator refers to jump from one knowledge unit to another knowledge unit, while path connection operator connects classes or instances belonging to the same knowledge unit. In the above equation, m ($m \geqslant 0$) and n ($n \geqslant 0$) are integers, and denote the number of paths to be connected by different path connection operator respectively. When $m = 0$, the combination operator becomes $(isa)^{n*}$. By analogy, when $n = 0$, the combination operator turns out to be a^{m*}.

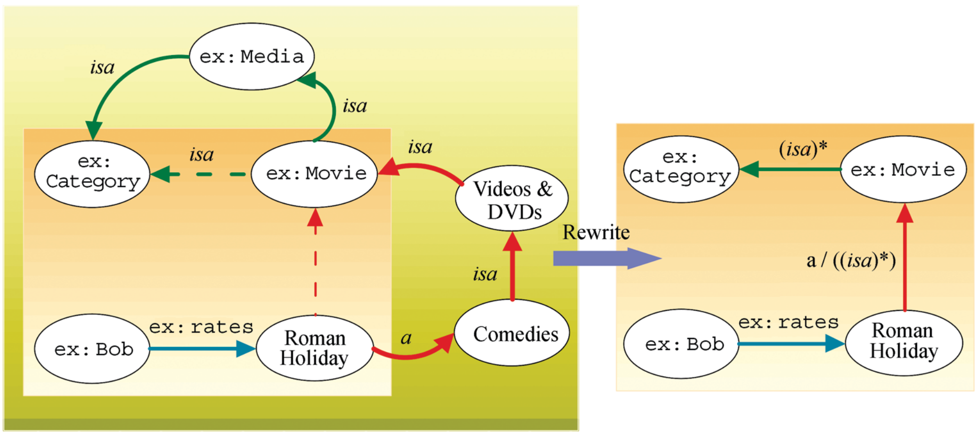

Fig.6. Rewrite query with path expressions.

The corresponding SPARQL query is as follows:

```
SELECT DISTINCT ?person ?category
WHERE  {?person ex:rates ?product
        ?product a/((isa)*) ?category
        ?category (isa)* ex:categary
       }.
```

By introducing Semantic Web technologies, machines can process the query with path expressions easily without human's effort. Therefore, we can obtain the query result directly and no need to care about the operations in underlying database.

7 Experiments

In this section, we perform experiments to show the effectiveness of the mechanism proposed in this article for trust mining. First, we describe the dataset used for experiments. Then, we present the experimental methods. Finally, we give the experimental results and discuss the practical value of our proposed method in real applications.

7.1 Dataset

The dataset used in this paper is obtained from the real-life online social network — Epinions. The data on Epinions can be categorized into two types: structured data and unstructured data. Structured data refers to product rating, review rating, belonging category and so on, while unstructured data refers to user-generated content such as users' review text, comments and self-introduction. Processing unstructured data requires techniques such as natural language processing and opinion mining, which is much more complicated. For simplicity, we mainly focus on structured data on Epinions in this paper. So, we extract four kinds of data as follows:

- *User Information*: user account, user's role;
- *Trust Relations*: explicit trust relationships from web of trust;
- *Review Information*: the author, product name, belonging category, and review rating;
- *Feedback Information*: review title, the user who offered feedback, and feedback rating, etc.

The first step of data collection is to get the user list. Since Epinions does not provide a list of all the online users, we obtain the data by crawling web of trust. We start crawling from an arbitrary top reviewer *bradshawcl* and follow both his *trusts* and *trusted-by* links to find other users. The crawling is in breadth first search and expands out from the staring point. Note that although we randomly choose a top reviewer as the starting point, the crawling expands out through the whole community. There are two reasons for us to choose one of the top reviewers as the starting point. First, Epinions does not provide a list of all the users, so the most convenient way to obtain the user list is to start from a top reviewer for crawling and follow both of his *trusts* and *trusted-by* links. Second, a top reviewer is more likely to have a set of high quality acquaintances, which makes it easier for us to mine useful trust relationships. In this way, we can obtain the user list and explicit trust relationships between users simultaneously.

We notice that a small percentage of users hide their web of trust from others to protect privacy. So we are not able to obtain their explicit trust relationships from publicly available data. Meanwhile, we also find that there is a certain percentage of inactive users on Epinions. These users have not logged on to Epinions for years and the information about them on the website is out of date. It is meaningless to reason about their interests and trust relationships. Therefore, we delete the above two kinds of users from the sample dataset.

After the user list is determined, we continue to crawl system-provided information, such as users' roles, product names, and belonging categories. We also extract data from user-generated content, such as review titles, product ratings, and feedback ratings. Note that, some of the top reviewers have written hundreds and thousands of reviews on Epinions. Therefore, there is a great deal of data that is hard to be processed. Meanwhile many previous reviews cannot reflect users' current status. Therefore we choose to crawl the latest 500 reviews (the total number of reviews written by some users is less than 500, then we crawl all their reviews).

7.2 Experimental Method

In this paper, we exploit Jena to perform semantic-based reasoning for mining trust relationships. Jena is an open source Java framework which provides rule-based reasoning functions for RDF and OWL, meanwhile has complete support for storage and persistence of knowledge base[29].

First, we utilize Jena to establish an RDF-based knowledge base. As illustrated above, we obtain related data of all the users in sample dataset. Then, we write a program to transform the above information into RDF representation and import those data into the knowledge base. Afterwards, we exploit Jena software development kit to perform reasoning in real application scenarios based on established domain ontology. The experimental results are provided below.

7.3 Experimental Results

In this subsection, we present three sets of experimental results obtained by using the reasoning mechanisms introduced in Section 5. The first one is to reason about users' interests, the second one is to discover implicit trust relationships, and the third one is to infer category-specific trust relationships. We will describe in more details respectively below.

7.3.1 Infer Users' Interests

In this experiment, we mainly consider the precision of reasoning results about users' interests. We define *precision* as the percentage of actual categories that are correctly identified by our proposed mechanism among all the discovered categories. Since we do not have a standard answer of users' interested categories (it requires a thorough investigation questionnaire of all the sample users about their interests), we find two persons who are experts at product review sharing to judge a user's interests according to his/her online archives. On each user's homepage, Epinions provides a user's written reviews, author popularity, feedback ratings, comments and self-introduction. Human experts can identify a user's interests based on the above system-provided and user-generated content. Usually, a user may have many interested categories. Human judges must read a great deal of content and tens to hundreds of reviews posted by a user to identify his/her interests. Therefore, it is not possible for human judges to do that for a large number of users, so we randomly select 100 users from our dataset as the sample for this experiment. We should also notice that the two expert judges are not part of our research team. They are independent consultants.

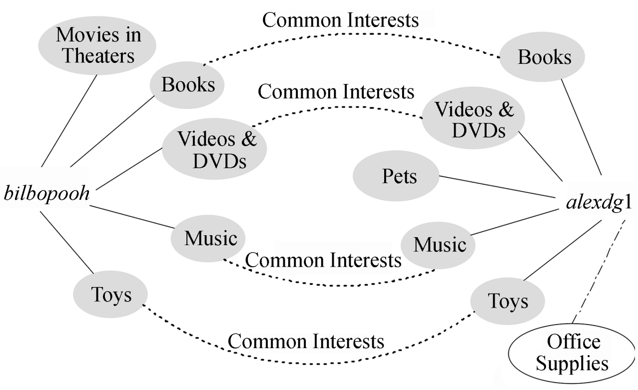

Fig.7. Example results of inferring users' interests.

Fig.7 shows an example of inferring users' interested categories. There are two Epinions users: *bilbopooh* and *alexdg*1. As can be seen from Fig.7, we reason that *bilbopooh* is interested in five categories: Movies in theaters, Books, Videos & DVDs, Music, and Toys. According to human experts' judgement, the above five categories are proven to be correct.

As for user *alexdg*1, the reasoning result turns out to be Books, Videos & DVDs, Pets, Music, Toys, and Office Supplies. According to judgement of experts, *alexdg*1 is not interested in Office Supplies. He just wrote several reviews on the stationeries he used in daily life. So, five out of six categories is regarded as correct.

In this way, we evaluate the correctness of all 100 users in the sample dataset. The average precision turns out to be 94.3%. This number shows that our reasoning mechanism is very effective in inferring users' interests automatically. Meanwhile, we can see from Fig.7, two users share quite a few common interests, and similar users are more likely to affect each other when making purchasing decisions. Therefore, determining users' interests is the first step of mining trust relationships and is also very important to targeted marketing and product promotion through online social networks.

Another advantage of our proposed method is that we are able to identify a user's interests which is not easy to be discovered. Take user 4-1-1 for example, he writes a large number of informative reviews on beers, wines and sprits, meanwhile he is well recognized in those fields on Epinions. However, the above categories are not shown in his author popularity list, nor in his self-introduction. As can be seen from Epinions, the topology of product categories is very complex. So, it is hard for human judges to correctly pinpoint the precise category of users' interests. They have to read a large number of reviews and comments. However, using our proposed method, we manage to correctly infer users' implicit interests without much human effort.

7.3.2 Infer Implicit Trust Relationships

Fig.8 shows the example results of inferring implicit trust relationships for Epinions users. The x-axis denotes different users, while the y-axis denotes the percentage of trust relationships being identified using different methods. Let W_{u_i} denote the set of explicit trust relationships that u_i has in web of trust. Let H_{u_i} denote the set of $u_i's$ trust relationships being discovered by our trust mining method. Then, we use $M_{u_i} = W_{u_i} \cap H_{u_i}$ to represent the common trust relationships that show up in both sets W_{u_i} and H_{u_i}. And let S_{u_i} denote the set of all the trust relationships being discovered for u_i, that is $S_{u_i} = W_{u_i} \cup H_{u_i}$. Then, let I_{u_i} denote the set of implicit trust relationships that are mined by our proposed method only, that is $I_{u_i} = H_{u_i} - M_{u_i}$. Notice that, the trust relationships that are mined by our method all have the category information. Then, we remove M_{u_i} from W_{u_i} and the remaining trust relationships in W_{u_i} are generalized trust relationships, denoted as set G_{u_i} and $G_{u_i} = W_{u_i} - M_{u_i}$.

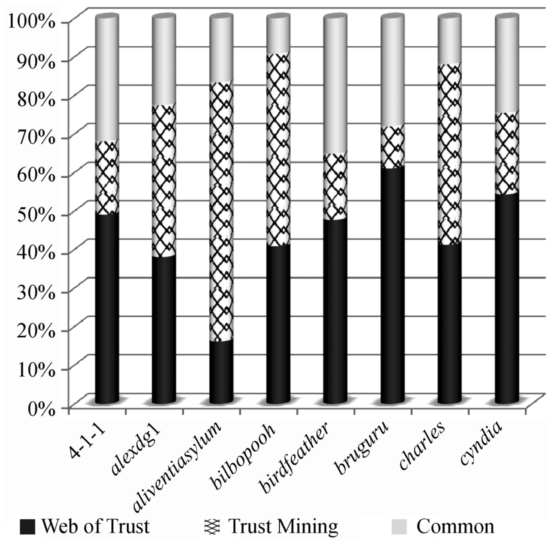

Fig.8. Example results of inferring implicit trust relationships.

Comparing our trust mining result set H_{u_i} with the web of trust set W_{u_i}, we find that the common set M_{u_i} accounts for about 15% to 40% of all the trust relationships being discovered, shown as the light grey column part in Fig.8.

As can be seen, there is also a certain percentage of trust relationships that only appear in the web of trust, that is G_{u_i} (shown as the black column part in Fig.8). We do not know the category information of those trust relationships between pairs of users. Some of those users are not active any longer, some of them do not pay attention to the trustee's reviews recently, while others just do not have any feedback behaviors on Epinions.

Meanwhile, we manage to identify a large number of implicit trust relationships which are shown as the grid column part in Fig.8. Those implicit trust relationships are submerged in large amount of user-generated content. Take user *patsyv* for example, she has read 491 out of 500 of top reviewer *byran_carey*'s reviews and all offered him positive feedback ratings (including "Helpful" and "Very Helpful"). However, she does not add *byran_carey* to her web of trust (*byran_carey* is not in *patsyv*'s *trusts list* until we submitted this paper. We are not sure whether she will add *byran_carey* to her web of trust in the future). That is, through the explicit trust relationships, we even do not know whether there exists a link between users *patsyv* and *byran_carey*. On Epinions, there is a certain percentage of users like *patsyv*. Without semantic-based reasoning functions, it is very difficult for us to identify those hidden and implicit trust relationships. If we only focus on explicit trust relationships in web of trust, those implicit trust relationships are totally ignored.

In Fig.8, we can see that we manage to obtain the category information for nearly half of all the trust relationships for sample users. For most users, the percentage exceeds 50%. For a small group of users, the figure can reach 85% (take user *aliventiasylum* for example). With category-specific trust relationships, we manage to conduct more efficient targeted marketing and product promotion through online social networks.

7.3.3 Infer Category-Specific Trust Relationships

As mentioned above, Epinions does not provide category information of trust relationships in web of trust. Using our trust reasoning mechanism, each identified trust relationship is associated with a category label. According to the label, we will know in which category that one user trusts in another. Then, we manage to carry out targeted marketing and product recommendation accordingly through online social networks rather than sending annoying spam or messages to users who have no interests in those fields.

Meanwhile, we can make use of category-specific trust relationships to determine influential reviewers in each category. Imagine that we want to convince a group of customers to purchase a product and our marketing budget only allows offering only a few free trials to potential customers. In order to achieve the maximized awareness of the product among customers, we should offer samples to influential people who may affect a large set of their friends, acquaintances or other online customers. Influence is also category-specific. Alice believes in John's point of view on computer hardware, but she does not appreciate his tastes of books. If we try to convince Alice to buy the book that John likes, it is probably just a waste of time and money. Under such circumstances, we can take advantage of category-specific trust relationships to identify influential users. We will illustrate in more details below using a real example on Epinions.

Fig.9 shows the influence quantification of all the top reviewers in *Book* category (the total number of

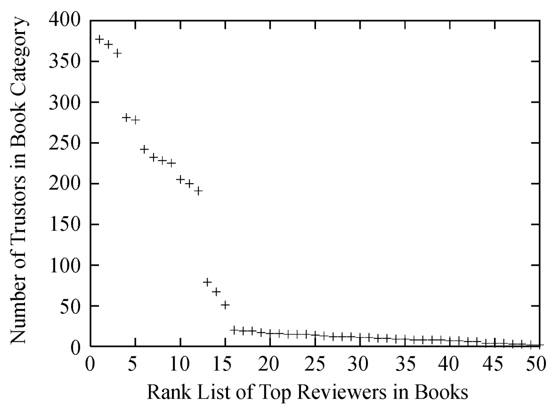

Fig.9. Influence of top reviewers in Book category.

top reviewers in this category is 50). We reason out the number of users who trust in a top reviewer in this field based on his/her latest 500 reviews. The greater the number, the more influence he/she may have in the online community. We take this number as a measurement criterion of a user's influence. In Fig.9, the x-axis shows the rank order of all the top reviewers in *Book* category, and y-axis shows the number of trustors each reviewer has. It is very clear to see that the influence of each reviewer differs greatly. These top reviewers can be divided into four groups: the first group has more than 360 trustors; the second group has about 190 to 280 trustors; the third group has about 50 to 80 trustors; while all the rest fall into the fourth category, the number of trustors that they have is less than 20.

Let R denote the set of all the trustors we obtained from our reasoning result. The total number of users in R is 577. Suppose that one top reviewer is influential enough to activate a user who trusts in him/her to make purchasing decision. In *Book* category, top reviewer *byran_carey* has the most trustors. He is able to activate nearly 65% of users in R based on our assumption. The second top reviewer is *popsrocks*, who is able to activate 371 users in all. Top reviewers *byran_carey* and *popsrocks* together can activate 482 users altogether, which accounts for 83.5% of users in R. If we add the third top reviewer *stephen_murray*, then 505 users in R are likely to be activated by them, which accounts for 87.5% of users in R. Through observation, there are a large number of users who trust the same group of top reviewers, therefore they are affected by a group of super users rather than only a single person. The influence effect is greatly strengthened under such circumstances. Based on our reasoning results and statistical data, companies can make marketing plans according to their budgets and requirements. For example, if we only have two free trials that can be offered to potential customers in *Book* category, then *byran_carey* and *popsrocks* are the two best choices to be made.

8 Conclusions and Future Work

In this paper, we present a method of mining trust relationships automatically from online social networks. The contribution of this article is multi-fold. First, we propose a semantic-based trust reasoning mechanism for trust mining in the online community. To the best of our knowledge, this is the first effort to propose the idea in literature. Second, we emphasize the category attribute of pairwise relationships and exploit Semantic Web technologies to describe category-specific trust relationships. Third, we adopt role-based and behavior-based reasoning functions to infer users' interests, implicit trust relationships and category-specific trust relationships. Fourth, we utilize path expressions to extend the reasoning rules, so that complex queries can be processed directly using derived knowledge in knowledge base without much human effort. The experimental results on real-life Epinions data show that our proposed method is very effective in mining trust relationships from online social networks. Meanwhile, our method has a wide application value in the realm of e-commerce, product recommendation, targeted marketing and product review, etc.

There is a lot of work to do in the future. First, we will make use of some prediction tasks, for example product recommendation, to provide more solid evaluations on the effectiveness of our proposed method. Second, we will make comparisons between typical statistical method and our semantic-based approach. Even more, we may combine the above two different methods to see whether we can achieve better mining results. Third, there is still a large amount of user-generated unstructured data available in the online community, which is very essential to more accurate and fine-grained trust evaluation. In the future, we will exploit the techniques such as opinion mining and sentimental analysis to conduct more comprehensive and deep trust mining. Since there is a large amount of data available in online social networks, we have very good platforms to conduct our further research.

References

[1] Guha R, Kumar R, Raghavan P, Tomkins A. Propagation of trust and distrust. In *Proc. the 13th Int. Conference on World Wide Web* (*WWW 2004*), New York, NY, USA, May 17-22, 2004, pp.403-412.

[2] Golbeck J, Parsia B, Hendler J A. Trust networks on the Semantic Web. In *Proc. the 7th Int. Workshop on Cooperative Intelligent Agents*, Helsinki, Finland, August 27-29, 2003, pp.238-249.

[3] Richardson M, Agrawal R, Domingos P. Trust management for the Semantic Web. In *Proc. the 2nd Int. Semantic Web Conference*, Sanibel Island, Florida, USA, October 20-23, 2003, pp.351-368.

[4] Massa P, Avesani P. Trust-aware collaborative filtering for recommender systems. In *Proc. Int. Conference on Cooperative Information Systems*, Larnaca, Cyprus, October 25-29, 2004, pp.492-508.

[5] Bizer C, Oldakowski R. Using context- and content-based trust policies on the Semantic Web. In *Proc. the 13th Int. conference on World Wide Web*, New York, NY, USA, May 17-22, 2004, pp.228-229.

[6] Tang J, Sun J, Wang C, Yang Z. Social influence analysis in large-scale networks. In *Proc. the 15th ACM SIGKDD Int. Conference on Knowledge Discovery and Data Mining* (*SIGKDD 2009*), Paris, France, June 28-July 1, 2009, pp.807-816.

[7] Zhang Y, Wang Z Q, Xia C L. Identifying key users for targeted marketing by mining online social network. In *Proc. the 24th Int. Conference on Advanced Information Networking and Applications Workshops*, Perth, Australia, April 20-23, 2010, pp.644-649.

[8] Pan W, Liu N N, Xiang E W, Yang Q. Transfer learning to predict missing ratings via heterogeneous user feedbacks. In *Proc. the 22nd Int. Joint Conference on Artificial Intelligence* (*IJCAI 2011*), Barcelona, Spain, July 16-22, 2011, pp.2318-2323.

[9] He Q, Pei J, Kifer D, Mitra P, Giles L. Context-aware citation recommendation. In *Proc. the 19th Int. Conference on World Wide Web*, Raleigh, North Carolina, USA, April 26-30, 2010, pp.421-430.

[10] Wang X J, Yu M, Zhang L, Cai R, Ma W Y. Argo: Intelligent advertising by mining a user's interest from his photo collections. In *Proc. the 3rd KDD Workshop on Data Mining and Audience Intelligence for Advertising*, Paris, France, June 28, 2009, pp.18-26.

[11] Tang J, Lou T C, Kleinberg J. Inferring social ties across heterogeneous networks. In *Proc. the 5th ACM Int. Conference on Web Search and Data Mining* (*WSDM 2012*), Seattle, Washington, February 8-12, 2012, pp.743-752.

[12] Tang W B, Zhuang H L, Tang J. Learning to infer social ties in large networks. In *Proc. ECML/PKDD 2011*, Athens, Greece, September 5-9, 2011, pp.381-397.

[13] Leskovec J, Huttenlocher D, Kleinberg J. Predicting positive and negative links in online social networks. In *Proc. the 19th International World Wide Web Conference* (*WWW 2010*), Raleigh, NC, USA, April 26-30, 2010, pp.641-650.

[14] Leskovec J, Huttenlocher D, Kleinberg J. Signed networks in social media. In *Proc. the 28th Int. Conference on Human Factors in Computing Systems* (*CHI 2010*), Atlanta, Georgia, USA, April 10-15, 2010, pp.1361-1370.

[15] Zolfaghar K, Aghaie A. Mining trust and distrust relationships in social web applications. In *Proc. the 2010 IEEE Int. Conference on Intelligent Computer Communication and Processing*, Cluj-Napoca, Romania, Aug. 26-28, 2010, pp.73-80.

[16] Nguyen V A, Lim E P, Jiang J, Sun A. To trust or not to trust? Predicting online trusts using trust antecedent framework. In *Proc. the 9th IEEE Int. Conference on Data Mining*, Miami, Florida, USA, Dec. 6-9, 2009, pp.896-901.

[17] Chua F C T, Lim E P. Trust network inference for online rating data using generative models. In *Proc. the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Washington, DC, USA, July 25-28, 2010, pp.889-898.

[18] Nguyen V A, E P Lim, Tan H H, Jiang J, Sun A. Do you trust to get trust? A study of trust reciprocity behaviors and reciprocal trust prediction. In *Proc. the SIAM Int. Conference on Data Mining*, Columbus, Ohio, USA, April 29-May 1, 2010, pp.72-83.

[19] Liu H F, Lim E P, Lauw H W, Le M T, Sun A X, Srivastava J, Kim Y A. Predicting trusts among users of online communities: An Epinions case study. In *Proc. the 9th ACM Conference on Electronic Commerce*, Chicago, Illinois, USA, July 8-12, 2008, pp.310-319.

[20] Ma N, Lim E P, Nguyen V A, Sun A X, Liu H F. Trust relationship prediction using online product review data. In *Proc. the 1st ACM Int. Workshop on Complex Networks Meet Information & Knowledge Management*, Hong Kong, China, November 2-6 2009, pp.47-54.

[21] Matsuo Y, Yamamoto H. Community gravity: Measuring bidirectional effects by trust and rating on online social networks. In *Proc. the 18th Int. Conference on World Wide Web*, Madrid, Spain, April 20-24, 2009, pp.751-760.

[22] Skopik F, Truong H L, Dustdar S. Trust and reputation mining in professional virtual communities. In *Proc. the 9th Int. Conference on Web Engineering*, San Sebastian, Spain, June 22-26, 2009, pp.76-90.

[23] Epinions. http://www.epinions.com, 2011.

[24] The Foaf Project. http://www.foaf-project.org/, 2011.

[25] SIOC core ontology specification. http://rdfs.org/sioc/spec/, 2011.

[26] GoodRelations: The Web vocabulary for e-commerce. http://www.heppnetz.de/projects/good relations/, 2011.

[27] OWL Web ontology language guide. http://www.w3.org/TR/owl-guide/, 2011.

[28] Defining n-ary relations on the Semantic Web: Use with individuals. http://www.w3.org/2001/sw/BestPractices/OEP/n-aryRelations-20040623/, 2011.

[29] Jena Semantic Web framework. http://jena.sourceforge.net/downloads.html, 2011.

Yu Zhang is an associate professor of Zhejiang Sci-Tech University, China. She received her Ph.D. degree from Zhejiang University in 2009. She is a member of CCF and ACM. Her current research interests include trust computing, Semantic Web and online social network analysis.

Tong Yu is a Ph.D. candidate in College of Computer Science of Zhejiang University, China. He received his M.S. degree from Zhejiang University in 2006. His current research interests include Semantic Web, data mining and biomedical informatics.