A Robust and Knot-Aware Trust-Based Reputation Model

Nurit Gal-Oz, Ehud Gudes and Danny Hendler

Abstract Virtual communities become more and more heterogeneous as their scale increases. This implies that, rather than being a single, homogeneous community, they become a collection of *knots* (or sub-communities) of users. For the computation of a member's reputation to be useful, the system must therefore identify the community knot to which this member belongs and to interpret its reputation data correctly. Unfortunately, to the best of our knowledge existing trust-based reputation models treat a community as a single entity and do not explicitly address this issue. In this paper, we introduce the *knot-aware* trust-based reputation model for large-scale virtual communities. We define a *knot* as a group of community members having overall "strong" trust relations between them. Different knots typically represent different view points and preferences. It is therefore plausible that the reputation of the same member in different knots assign may differ significantly. Using our knot-aware approach, we can deal with heterogeneous communities where a member's reputation may be distributed in a multi modal manner. As we show, an interesting and beneficial feature of our knot-aware model is that it naturally prevents malicious attempts to bias community members' reputation.

Nurit Gal-Oz
Deutsche Telekom Laboratories at Ben-Gurion University, Beer-Sheva, 84105, Israel, e-mail: galoz@cs.bgu.ac.il

Ehud Gudes
Deutsche Telekom Laboratories at Ben-Gurion University, Beer-Sheva, 84105, Israel, e-mail: ehud@cs.bgu.ac.il

Danny Hendler
Deutsche Telekom Laboratories at Ben-Gurion University, Beer-Sheva, 84105, Israel, e-mail: hendlerd@cs.bgu.ac.il

Please use the following format when citing this chapter:

Gal-Oz, N., Gudes, E. and Hendler, D., 2008, in IFIP International Federation for Information Processing, Volume 263; *Trust Management II*; Yücel Karabulut, John Mitchell, Peter Herrmann, Christian Damsgaard Jensen; (Boston: Springer), pp. 167–182.

1 Introduction

The Internet has enabled the creation of virtual worlds and communities, where user interactions imitate and, to some extent, even replace the more traditional "real-life" equivalents on a larger scale. The existence of easily accessible virtual communities makes it both possible and legitimate to communicate with total strangers. We can now anonymously interact with other virtual community members whom we do not really know in ways that are, in general, not possible in the "real world".

Many of our real-life decisions, such as, e.g., which book to buy or which physician to consult with, are based on information that we collect based on our social interactions. Our decisions are often based on our own direct experience, but when direct experience is lacking, we have to rely on the opinions of people we know. In the latter case, the weight we assign to these opinions is directly correlated to the extent by which we trust these people. This decision-making process is obviously individual and is often based on hunches and intuitions: we are not always fully aware of why we eventually decide as we do. While representing the data used for such decisions electronically is relatively straightforward, it is much harder to imitate human decision making processes once that information is available. This is a key difficulty in devising effective virtual communities. Another difficulty is that virtual communities expose their members to new types of fraud and deception since impersonation is much easier in virtual communities than in the "real-world".

Trust and reputation systems are considered key enablers of virtual communities, especially communities of strangers, where users are not required to reveal their real identities and use nicknames or pseudonyms instead. These systems support the accumulation of member reputation information and leverage this information to increase the likelihood of successful member interactions and to better protect the community from fraudulent members. Unlike cryptographic-based security mechanisms to which a user typically must be explicitly aware, trust and reputation systems are often considered to be a *soft security* mechanism [15]: "Soft security accepts and even expects that there might be unwanted intruders in the system. The idea is to identify them and prevent them from harming the other actors".

As the scale of virtual communities continues to increase, they become more and more heterogeneous. This implies that, rather than being a single, homogenous community, they become a collection of loosely-coupled *knots* (i.e. sub-communities) of users. We define a *knot* as a group of community members having overall "strong" trust relations between themselves (see Section 3.3 for a formal definition). Typically, members belonging to the same knot are more likely to have similar viewpoints and preferences as compared to members that belong to different knots.

In this paper, we present the knot-aware trust-based reputation model. We model virtual communities of strangers, where members seek services or expert advice from other members. Two key examples of such communities are eBay [1] and Experts-Exchange [2]. A major implication of the fact that community members do not necessarily expose their real identity is that trust must rely on reputation accumulated by member ratings and cannot be derived from "real-world" social familiarity. The assumption underlying our knot-aware model is that "less is more":

the use of relatively small, but carefully selected, subsets of the overall community's reputation data yields better results than those represented by the full data set. The application of this principle is done by (1) partitioning the community into knots of members who rated other members in a similar manner, and (2) assigning higher weight to the reputation information of intra-knot members than that of members outside the knot. Since members are primarily influenced by members that shared their preferences in the past, a useful feature of our model is that it naturally prevents malicious attempts to bias community members' decisions. Another advantage is that smaller sub-communities, whose viewpoints differ from the overall community average, can maintain their distinctive preferences without having their opinions "diluted' by those of the majority of users outside their knot.

We have implemented an algorithm that adheres to our knot-aware model and conducted experiments to evaluate it by using the publicly available database of the MovieLens [3] movie recommendations community. Our preliminary evaluation results establish that our knot-aware approach improves system predictions as compared with non-knot aware trust computation.

The rest of the paper is organized as follows. Section 2 provides an overview of the related work. In section 3, we formally define our knots-aware model. Simulation results are presented in Section 4. We conclude by discussing future research directions in Section 5.

2 Related Work

Trust and reputation models for virtual communities are gaining increasing research attention. Several such models have been proposed over the last decade. These models differ mainly in how they define trust and reputation, in their assumptions on how the system obtains data on trust relationships, in how they compute new trust relationships based on existing data, and in the set of trust *attributes* (i.e., different trust dimensions) used by them.

Abdul-Rahman and Hailes [4] proposed a model in which the trust attributed to one member by another falls into one of four levels: *VeryTrustworthy*, *Trustworthy*, *Untrustworthy* and *VeryUntrustworthy*. As they mention, trust in their model is not transitive, hence a member only relies on recommenders with which it has "personal experience". *Ratings* are the smallest building block on which trust and reputation models are built. [4] use the *semantic distance* of ratings to identify recommenders that are *over-raters* or *under-raters* and adjust their recommendation accordingly. The trust in a recommender is calculated after a transaction between two members is completed, by evaluating the distance between the recommendation and the actual experience. In computing a recommended trust value (reputation), the model gives higher (fixed) weight to information coming from members with a more similar point of view.

Yu and Singh [18] use Dempster-Shafer's theory of evidence to represent and propagate the ratings given by agents to others. In their model, a user rates an agent

upon each transaction as either trusted or untrusted. These ratings are collected and used as evidence to determine the user's level of trust in that agent. Transitive trust-chains of predefined length are generated, in order to identify the *witnesses* of the target agent, and information from different witnesses is aggregated using Dempster's rule of combination.

The Beta reputation system introduced by Josang [8] is based on using beta probability density functions to combine feedback and derive reputation ratings. In their model, the authors use a *forgetting factor* to overcome the growing irrelevance of old feedback due to changing behavior of agents over time. However it only considers the order of experiences but does not factor in the elapsed time. That is, the one-before-last feedback is "forgotten" to the same degree regardless of whether it was given yesterday or a year ago. An aging factor is used also in Kinateder [11] to update trust. This factor determines how fast new experiences change the trust compared to previous experiences, regardless of the time distance.

Some of the prior art works do not distinguish between trust in a target agent (service provider) and trust in a recommender [10, 19]. These assume that a good service provider will be a reliable witness as well. While this may be true for some systems (e.g., in file sharing systems), it is not necessarily the case for other communities. In other works [4, 11], the requesting user is required to rate the recommendation she got after the transaction took place. In their paper [11], the authors argue that in ratings that consist of multiple attributes, the process of collecting user feedback cannot be automated and every single recommendation must be rated. This requirement is highly challenging, since soliciting member feedback on transactions is a major challenge in reputation systems; receiving feedback on each and every recommendation might not be feasible. TrustBAC [5] relies on the agents' ability to provide recommendations but it is not clear how these recommendations are obtained. TrustBAC is a trust-based access control model that is somewhat different from other models of trust and reputation, as it actually extends the conventional role-based access control model with the notion of trust levels. The trust an agent puts in another is represented by a vector with three components: (1) the *experience* with the target agent, (2) *knowledge*, specifying whether the experience is direct or indirect, and (3) *recommendation* - the cumulative effect of all recommendations from different sources.

In Jimminy [12], the authors present an honesty assessment algorithm and a reward model for encouraging honest ratings. Their computation of honesty is based on the probability distribution of all ratings available for a subject. Their model assumes there is a correlation between the extent by which a member's ratings disagree with the ratings of others and the probability that the member is dishonest.

In the group based reputation system proposed in [17], groups are constructed according to users shared interests and trust is calculated based on user ratings between peers, between groups and peers, and between groups. A similarity based clustering is used to identify the set of trustworthy raters within a group and filter unfair ratings, however this approach is aimed at certain types of applications (e.g. file sharing).

A review on trust and reputation systems is provided by Josang et al in [9]. Their review discusses the semantics of the trust and reputation concepts and the relations between them, and suggest a classification of trust and reputation measures according to two dimensions: *specific* vs. *general*, and *subjective* vs. *objective*. The authors also provide an overview of reputation computation models and existing applications of online reputation systems. Sabater et al [16] also presents an overview of several computational trust and reputation models that have been implemented and classify them according to a few criteria, such as the source of the information used, the assumptions made on agents' behavior, the visibility of trust, and whether reputation is considered as a personal/subjective property or as a global property.

3 Trust Based Reputation Model

The knots model is composed of three separate modules: the *member trust inference module*, the *knots construction module*, and the *reputation computation module*. The member trust inference module identifies trust relations among members; the knots construction module utilizes these relations to to generate trust knots; the reputation computation module computes local reputations within knots and global reputations for the community as a whole. We describe the key responsibilities of these modules later in this section.

3.1 Trust and Reputation

As mentioned before, our model is designed for communities in which members typically do not reveal their real identities. We consider communities in which experts in specific fields offer their advice and consulting services to community members seeking such services. A community consists of *members*, all of which may participate in community activities, such as searching for an expert, interacting with an expert, and sharing recommendations about experts with other members. *Experts* are a subclass of *members*. Experts may provide professional services and advice to members. Although experts are also members, in what follows we regard experts and members as two disjoint sets for presentation simplicity.

The literature proposes several definitions of the term *trust*. We adopt Mui's definition [14] because it emphasizes the two major aspects of trust - the subjectivity of trust and the importance of accumulated experience - and extend it to capture a third aspect: the transitivity of trust. When searching for an expert, a member takes into consideration the extent to which the members she trusts trust relevant experts.

Definition 1. *Trust* is a subjective expectation an agent has about another's future behavior based on the history of that agent, and other reliable agents' encounters with that other agent.

As suggested in [5], we use the term trust in two different contexts.

Definition 2. *TrustMember* (TM) is Trust in the context of recommendations. More specifically, it is a trust value that quantifies the extent by which one member relies on another member to rate experts "correctly". *TrustExpert* (TE) is Trust in the context of experts. More specifically, it is a trust value that quantifies the extent by which a member relies on an expert to successfully provide the service it requires.

Reputation systems suggest various metrics for calculating reputations. Most of them compile an aggregated "general opinion" of all recommending members. Some of these systems disregard ratings that are too far from the popular rating score or even consider them as malicious [12]. We maintain that it is possible for the same experience to be perceived differently by different people. For example, the same PC expert may get excellent ratings from inexperienced PC users and very poor ratings from highly experienced PC user. When there is no clear normal distribution of the rating scores given, we cannot ignore the opinion of a minority or even try to aggregate the scores to one measure that will surely result in loss of valuable information.

We define reputation according to [14] and add the word "aggregated" to emphasize the fact that reputation is an aggregated quantity. This definition captures the time-dependency as well as the subjective nature of reputation.

Definition 3. *Reputation* is the aggregated perception that an agent creates through past actions about its intentions and norms.

3.2 Member Similarity and Trust Sets

Member ratings are the foundation of most trust and reputation models. *Rating* is a member's evaluation of the quality of a transaction after its completion. There are various aspects on which one can evaluate a transaction. A dimension based rating is a vector of values in the range [0,1], each representing the degree to which a member is pleased with the transaction from a certain aspect. While some of the dimensions depend on the type of transaction performed, others, such as, e.g., promptness or politeness, are more general and may be considered universal dimensions. Our model views a ratings as vectors of dimension weights with total weights sum of 1.

Definition 4. *Rating Similarity* quantifies the similarity between two comparable ratings (i.e., ratings given on the same expert in the same context within some limited time interval). The rating similarity of two ratings is defined as the Euclidean distance between the two rating vectors.

In trust-based reputation models of communities of strangers, such as [4, 10, 5], the trust a member has in another is based on how the trusting member rates the trusted party upon the completion of a transaction. In contrast, in models of communities in which members *are* familiar with each other, trust is based on personal acquaintance.

No explicit recommendations on experts are given by members in our model. Instead, we use the notion of *Member Similarity* (formally defined shortly) to calculate implicit trust among members. Let us elaborate on this idea. We observe that, in communities were explicit recommendations are provided, a recommendation given by one member to the other may be viewed as a transaction between the two members; rating the recommendation is a statement of how valuable it was. One way a member can evaluate the quality of recommendations provided by another member is to compare that member's recommendations with her own experience. In our model, member Similarity quantifies the extent of similarity between the ratings given by two members to the same experts. If the ratings of two members are very similar, this may be viewed as good ratings of the implicit recommendations given by each of them to the other. Hence, the influence of member A's ratings on the results of an expert-search done by member B increases as much as the member Similarity between A and B is higher.

Technically, member Similarity is a value in $[0,1]$, quantifying the extent to which two members are considered similar based on their common experience. Member Similarity may change over time and should thus be calculated based on all comparable ratings while assigning more weight to more recent ones. Our model takes time into account, by splitting history into a set of *time intervals*, each of which is assigned a different weight. This set may change over time. In any instance of time, the sum of interval weights is 1. Typically, more recent time intervals will be assigned higher weights as in [5]. In any instance of time, we only compare two ratings if they belong to the same time interval. The length of a time interval, the number of relevant time intervals and the weight assigned to each time interval are parameters that can be set according to the nature and maturity of the community. Formally, we define member Similarity as the opposite of member Difference, defined in the following.

Definition 5. Let $R^t(A,x)$ denote a rating vector given by member A at time t with respect to expert x. Let $R^{T_i}(A,x)$ denote the average (vector) of all ratings given by A with respect to x in time interval T_i. If there are no such ratings, then $R^{T_i}(A,x)$ is defined as the zero vector. Let X be a set of experts and let $TI \subseteq \{T_1,..,T_i\}$ be the set of time intervals for which comparable ratings with respect to the set of experts X were found. Let W_{T_i} denote the weight of interval T_i.

Then the *Difference* between two users in time $t \in T_i$ is given by:

$$Difference^t(A,B) = \sum_{T_j \in TI} W_{T_j} \cdot Average_{x \in X}(Distance(R^{T_j}(A,x), R^{T_j}(B,x)))$$

Definition 6. The Similarity of two members A and B at time t is defined by:

$$SIM^t(A,B) = 1 - (Difference^t(A,B))$$

Next, we define the *confidence* of member A in member B, which computes the significance of the transactions between A and B relative to the overall transactions in which A was involved.

Definition 7. Let $n(A,B)$ denote the number of ratings used to compare between A and B, let $N_A = \{n(A,B)|B \textit{ has comparable ratings with } A\}$. Then the confidence of A in the similarity with B is defined as:

$$SConf_{\beta}(A,B) = (\beta + (1-\beta) \cdot CDF(N_A, n_B).$$

Where CDF is the Cumulative Distribution Function. This function describes the statistical distribution of the number of comparable ratings A has with any other member. $CDF(N_A, n_B)$ returns for n_B in the sample N_A the probability of receiving this outcome or a lower one. β is the maximum Trust we allow A to have in B given perfect similarity $Sim(A,B) = 1$ and minimum experience; $(1-\beta)$ gives weight to previous experience.

Definition 8. The TrustMember function of A in B, denoted by $TM(A,B)$, quantifies the extent by which member A relies on the ratings of member B. It is defined as follows:

$$TM^t(A,B) = SIM^t(A,B) \cdot SConf_{\beta}(A,B).$$

The *Trust Set* of a member at some point in time is the subset of community members she trusts above some level. The idea behind the Trust Set concept is to allow every member to rely on members that provided ratings similar up to some degree α. The benefit of using Trust-Sets is in limiting the recommendation process to a smaller group of members that are better qualified for the task, while increasing the chance of getting more accurate results.

Definition 9. An α ***- Trust Set*** of member A denotes the set of all members whom A trusts with level α or more. The value of the α parameter depends on the domain and maturity of the community. We therefore use the term α-mature community to describe a community that requires α as the minimum value considered as trust. The α - *Trust Set* of member A at time t, denoted $TrustSet^t_{\alpha}(A)$, is defined as follows:

$$TrustSet^t_{\alpha}(A) = \{B|TM^t(A,B) \geq \alpha\}$$

3.3 Knots

A *knot* is a subset of community members identified as having overall strong trust relations among themselves. Every knot member should trust any other knot member (either directly or indirectly) above some threshold parameter. Two members A and B belong to the same knot if A has high enough direct trust in B (this implies that B is in A's trust set) or if A has high enough indirect trust in B (e.g., if A trusts C and C trusts B and by applying the transitive property of trust we conclude that A trusts B), and vice versa. knots are an extension of trust sets to create groups of members that can rely on each other's recommendations even if they did not rate the same experts. As will be discussed in section 3.5, knots have the ability of reducing

the risk of relying on dishonest or biased recommendations, since the members that provide these recommendations are identified and excluded from the knot.

Instead of using trust propagation as in [5, 10, 7], we use the transitivity property to make sure there exists a predefined level of propagated trust among a knot's members. Highly trusted members are identified within each knot. Once a member A is added to a knot, it relies on the recommendations of these members even if A itself does not have high trust in them (either directly or indirectly).

The problem of identifying knots in a trust network is modeled as a graph clustering (GC) problem, where vertices correspond to individual items and edges describe relationships. Under this interpretation, a community is represented by a directed graph $G = (V, E)$, in which vertices represent members and edges represent the trust relations between the members represented by their end-point vertices. The weight on a directed edge from A to B is the level of trust A has in B and is computed by $TM^t(A, B)$. Since our model is time-dependant, this graph changes over time.

We define the edges with weight at least α as *Positive Edges*, and all other edges as *Negative Edges*. We use clustering algorithm to partition the graph into a collection of subgraphs that represent knots. More formally, we define a knot as a cluster of vertices, from each of which there exists a path of positive edges of length k or less (for some system parameter k) to any other, and in which the number of negative edges does not exceed some system threshold parameter (possibly zero).

We use an algorithm based on the notion of *distance-k cliques* as the initial clustering algorithm. A sub-graph is distance-k clique, if any two vertices in it are connected by a path of length k or less. In their paper [6], the authors present a heuristic algorithm for graph clustering using Distance-k cliques. Their work refers to an unweighted and undirected graph. We extend their algorithm to use edge weights as additional criteria.

The value of parameter α is a significant factor in our model. If the group of "negative" edges is relatively small, a higher value of α may be used for obtaining smaller sub-graphs with stronger connectivity. In the opposite case, if there are too many "negative" edges, resulting in a partition into many small sub-graphs, we may lower the value of α for obtaining bigger knots. If lowering the level below some predefined level of α is required in order to achieve the desired number of components, this indicates that the community is immature (i.e. not enough data is available) and no valuable partition can be obtained at this stage.

Knots should be constantly updated in order to identify un-trusted members and add new trusted candidates. New members are added to a knot if they are found to significantly contribute to it. We define the ***strength*** of an α-knot as the total weight of all edges in the sub-graph representing this knot. Positive edges contribute positive strength while negative edges contribute negative weight, proportional to their distance from the α level of trust. Knot strength is used for computing the global expert reputation, as discussed in section 3.4.2. For each vertex, we compute the value of a *contribution function* that measures the strength a vertex adds to the knot. This value is based on the ratio between the number of negative and positive edges connected to the vertex. The contribution function is used in the knot update process. The update process is triggered by the transactions performed in the system

as follows. A transaction between member v and expert x, after which v rated x, updates the values of $TrustMember(v,u)$ and $TrustMember(u,v)$ for all members u who rated x in the current time interval (T_{now}). The value of $TrustSet^{t_{now}}(v)$ is also updated.

If v is not yet a member of a knot but has a trust set, it is added to the set of *un-clustered members*. If v is a member of a knot and its trust relations with members of the knot were modified, then this knot is reevaluated for checking whether there is a decrease in strength. If this is the case, then members with negative contribution may be removed from the knot and added to the set of un-clustered members. Finally, there is an attempt to add un-clustered members to knots to which they have the maximum positive contribution. A detailed description of the update algorithm is beyond the scope of this extended abstract and will be provided in the full paper.

3.4 Calculating Trust and Reputation

3.4.1 Reputation of members and experts

Local reputation is the reputation obtained based on information collected by the knot members, while global reputation is based on the complete data collected from the community. Formal definitions follow.

Definition 10. A *member's local reputation* is defined as the normalized sum of weights of the incoming edges of the vertex representing that member. Let $G_{KN_\alpha} = (V_{KN_\alpha}, E_{KN_\alpha})$ be the graph representing a knot of level α. Then the local reputation of member v is the relative portion of trust v gains in the knot and is defined as follows:

$$MLR^t(G_{KN_\alpha}, v) = \frac{\sum_{u \in V_{KN_\alpha}} TM^t(u,v)}{\sum_{(\forall u, k \in V_{KN_\alpha})} TM^t(u,k)}$$

Expert local reputation expresses the reputation of the expert within the knot. It is computed as the normalized weighted mean of all the ratings of the expert, where the weights are the local reputations of the rating members. This implies that members with higher local reputation contribute more to the final score of an expert. This is formalized by the following definition.

Definition 11. Let $TI \subseteq \{T_1, .., T_i\}$ be the set of time intervals for which member A rated expert x. Also let W_{T_j} denote the normalized weight of each time interval $T_j \in TI$ and let $DTE^t(A,x)$ denote the *direct trust A* has in expert x, calculated as the average of all ratings A gave x, weighted according to time interval weights, as follows:.

$$DTE^t(A,x) = \sum_{T_j \in TI} W_{T_j} \cdot R^{T_j}(A,x)$$

Then the local reputation of an expert x is defined as follows:

$$ELR^t(G_{KN_\alpha},x) = \frac{\sum_{(\forall v \in V_{KN_\alpha}, DTE^t(v,x)\neq\perp)} DTE^t(v,x)\cdot MLR^t(G_{KN_\alpha},v)}{\sum_{(\forall v \in V_{KN_\alpha}, DTE^t(v,x)\neq\perp)} MLR^t(G_{KN_\alpha},v)}$$

Global reputation is calculated based on all the ratings given to an expert within the community. In this computation, we take into account the local reputation as calculated within all knots, while the relative weight of each knot is a function of its strength within the community and the number of ratings used to calculate the local reputation of the expert within the knot. The *strength* of a knot containing a single member is defined as 0 and so it does not influence expert global reputation.

Using this method, vertices representing members that are knots of strength 0 (implying that no other member has trust in them) have no impact on the global reputation of experts. The real life meaning of this is that ratings given by new members will be disregarded until these members gain trust within the community. The global reputation of a new expert can be calculated after her first transaction, based on a rating provided by a single member, on condition that the member belongs to a knot (i.e., is she is trusted by a group of members). In this way, our model can prevent attempts by malicious experts to gain community reputation through friends that register as new members for that purpose.

As can bee seen from the above, an expert's reputation is composed of multiple scores. The expert's global reputation provides some general measure that assigns more weights to the ratings of trusted members and strong knots. For a node that belongs to a specific knot, however, the expert's local reputation within the knot is more useful.

3.4.2 Computing Expert Trust

TrustExpert - the extent by which a member trusts an expert - is used to assist a member in deciding with which expert to consult. We first consider the direct experience the member has with the expert. If there is no relevant direct experience, we turn to the members trust set. In case there is no information with respect to the expert there, we take the expert's local reputation within the knot of the member as the member's trust. Finally, if there is no local reputation for that expert within the knot, we turn to the whole community and use the global reputation score (see figure 1). A formal definition follows.

Definition 12. Let A and x respectively denote a member and an expert. Then the trust assigned by A to x in time t, denoted $TrustExpert^t(A,x)$, is defined as follows.

If A has direct (first hand) experience with x in the time interval [t-Tmax, t]:

$$TrustExpert^t(A,x) = TE^t(A,x) = DTE^t(A,x)$$

Else, if members of A's trust set have direct experience with x in $[t-Tmax,t]$:

$$TE^t(A,x) = \frac{\sum_{B\in TrustSet^t(A), DTE^t(B,x)\neq\perp} DTE^t(B,x)\cdot TM^t(A,B)}{\sum_{B\in TrustSet^t(A), DTE^t(B,x)\neq\perp} TM^t(A,B)}$$

Else, if no recent ratings exists within A's trust set w.r.t. x, or if A's trust set is smaller than some system threshold paramter:

$$TE^t(A,x) = ELR^t(G_{KN_\alpha(A)},x)$$

Finally, if A has no input from a trust set or a knot:

$$TE^t(A,x) = ExpertGlobalReputation^t(G_{community},x) = EGR^t(G_{community},x)$$

$$EGR^t(G_{community},x) = \frac{\sum_{i\in S} ELR^t(KN_i,x)\cdot Strength(KN_i)\cdot RatingPower(KN_i,x)}{\sum_{i\in S} Strength(KN_i)\cdot RatingPower(KN_i,x)}$$

Where S is the set of all knots of level α in which members rated x. $RatingPower(KN_i,x)$ is the number of ratings used to calculate the ExpertLocalReputation of x within knot KNi.

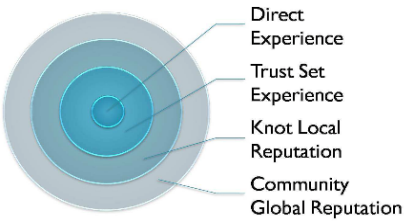


Fig. 1 Obtaining Trust in Expert

3.5 Resistance to Fraud and Malicious Member Behavior

One of the major benefits of the proposed model is its built-in ability to handle adversaries. Consider the two primary types of adversaries described by [13]: *selfish peers* and *malicious peers*. Selfish peers use the services of the community but do not contribute to the knowledge accumulated within it. The incentive provided by our model for not being selfish is that this allows a member to obtain more accurate information from members she trusts. Without providing ratings, a member cannot have a trust set nor can she belong to a knot (due to lack of trust in her). Thus, selfish members can only use the global and less accurate reputation data. In order to get the most out of the community services, one must provide ratings.

Consider a malicious member who wishes to harm the community (decrease the accuracy of the accumulated knowledge) by attempting to bias community ratings and members' reputation. Assume first that this member belongs to a knot. In order to join a knot, one must be trusted by a group of members. Since our model uses similarity as a measure of trust, if the malicious member's ratings deviate significantly from those of other knot members he is eventually removed from the knot. Thus, dishonest ratings will affect the knot for only a short period of time. Assume

that the malicious member does not belong to any knot.In this case, the malicious member has no adverse effect on any community member, as the global reputation computation ignores ratings provided by her.

4 Model Evaluation

Our experiments are based on a simulation of our model using data from MovieLens [3]. The MovieLens data provided by the GroupLens project consists of over a million movie ratings submitted of 3883 movies by 6040 users. In reputation systems, all participants -both members and experts - are dynamic entities that may change over time. In the Movielens web-site, however, a movie is rated only once by a user. In the absence of a more suitable data-set, we used the MovieLens data for evaluating some aspects of our model. We plan to conduct a more comprehensive evaluation of our model once we are able to obtain a ratings database that better suites our model.

In our simulations, movies play the role of experts. We analyzed the data in order to understand the impact of the β parameter on the selection of the α parameter for trust sets. In addition, we simulated the trust a member has in an expert based on her own experience, her trust set and her knot and demonstrated the benefits of using a relatively small set of members. The MovieLens rating consists of a single integer in the range $[1,5]$. We normalized this range to $[0,1]$ by linear scaling. In a 5 values rating scale, 20% error is measured for a one level difference from the member rating value. This explains the relatively high percentage error.

Prior to conducting our tests, a set of 20 movies was selected randomly and the ratings on these movies were excluded from the complete set of ratings. Three different values of the β parameter were examined: 0.6, 0.7 and 0.8. Trust sets of α level 0.7, 0.8 and 0.9 were produced for each of the three values of β. We analyzed the trust set for 150 members. The size of the trust set increased as we increased the β parameter. Increasing β results in assigning more weight to similarity and less weight to the experience factor. The reason for this is the large amount of experience in this community and the high variability of users' mutual experiences. On the other hand, the size of the TrustSets decreased as we required a higher level of trust by raising the α parameter value, as expected. The number of trust relations of level $\alpha = 0.9$ was about 2% of the number of trust relations of level 0.8 and about 0.5 percent of the number of trust relations of level 0.7. TrustSets of level 0.9 could not be produced for some of the members or were too small (5 members in average). A very large number of trust relations of level 0.7 resulted in very large trust sets (1400 members in average). For our evaluation, we chose to work with $\beta = 0.8$ and $\alpha = 0.8$, producing trust sets of 90 members on average. Trust sets corresponding to $\alpha = 0.7$ were used for comparison.

The set of tested members was constructed using three different types of users:low-raters, high-raters, and average-raters. The low-raters, high-raters, and average-raters are members that have low, high and average rating averages, respectively,

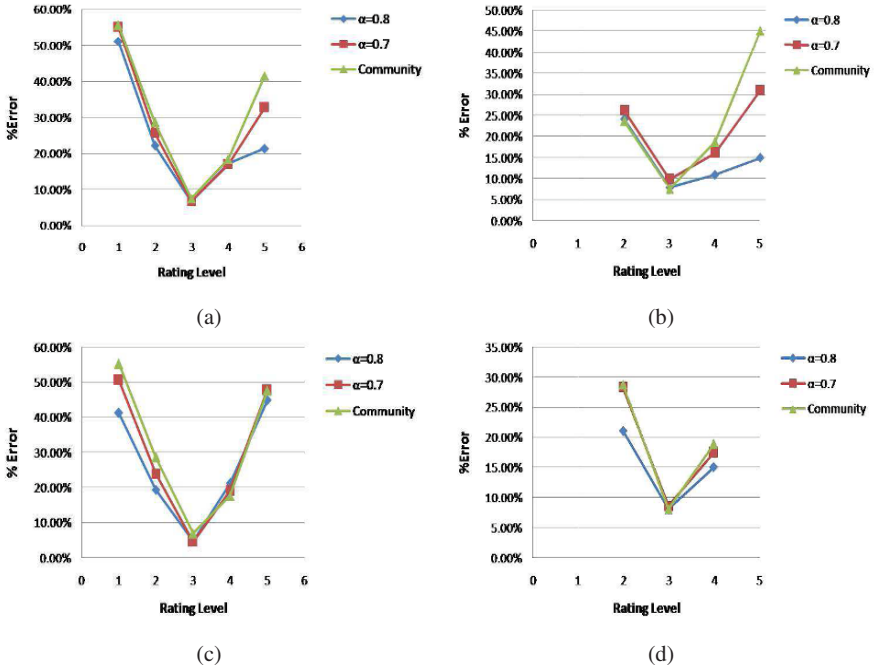
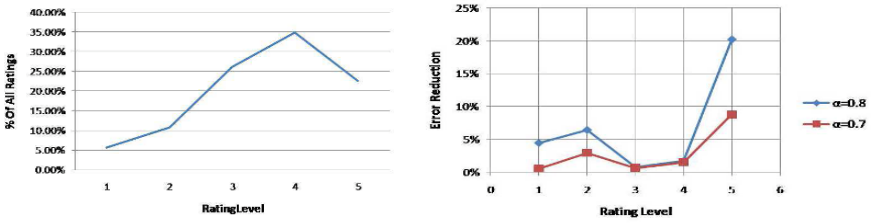


Fig. 2 Trust Sets Rating Error by member type: (a) All Members, (b) High Raters, (c) Low Raters, (d) Average Raters

and standard deviation smaller than 1. Our first goal was to evaluate the quality of the rating score derived by different levels of α trust sets. For each of the three types of users, we show the difference between the predicted rating of the trust set and the actual user rating on the pre selected movies. We found that, in general, trust sets of $\alpha = 0.8$ predict ratings better than trust sets of $\alpha = 0.7$ and that trust sets of $\alpha = 0.7$ predict ratings better than the overall community average. The average size of the $\alpha = 0.8$ trust sets was 75 while the average size of the $\alpha = 0.7$ trust sets was approximately 1400 (ranging from 261 to 2288). In addition, both trust sets yield better results than the average ratings of the overall community.

Figure 2(a) shows the average error-percentage for each of the 5 levels of ratings. It can be seen that, for ratings with high value, the number of errors is significantly reduced by the $\alpha = 0.8$ trust sets: it is reduced by 24% as compared with the average and by 14% as compared with $\alpha = 0.7$. For average-grade ratings, the improvement is less significant: more than 3% improvement for 4-ratings and less than 1% for 3-ratings as compared with the community average. For 1- and 2-ratings, the improvement was 5% and 6%, respectively. Splitting the data to the different groups of raters shows that, for the set of high raters, results improved as the ratings value increased: 26% better for ratings of 5 and 13% for ratings of 4, as compared with the community average. In the low rater set, we observe a similar phenomenon: 9%



(a) (b)

Fig. 3 (a) Distribution of ratings, (b) Error reduction by rating value

better for ratings of 2 and 13% better for ratings of 1. In the average raters set, the results are similar to the community average results: ratings of 5 are predicted 12% more accurately than the average community ratings. The other levels of rating are improved by only 1%-4%. In Figure 2(b),(c) and (d), we show the results for different types of users. The fact that low ratings are relatively rare in the data-set explains why the improvement is bigger on the high end of the rating scale than on the low end: only 5% of the ratings are of value 1 and approximately 10% of of value 2, while over 22% of the ratings are of value 5, as shown in figure 3(a). Figure 3(b) shows the average error reduction compared to the community average for each of the 5 levels of ratings.

A second goal of our experiments was to obtain the results for knots. The K parameter of clique distance was set to 6. We have examined three different types of knots, created around low raters, average raters and high raters. These consisted of 52, 101, and 119 users, respectively. The results for knots had correlation with the results for trust sets. In the high raters knot, the high ratings were predicted more accurately (around 9% improvement for ratings of 5 and 4% for ratings of 4). In the low raters knot, the low ratings were predicted with better accuracy (around 8% for ratings of 1 and 2).

5 Summary

In this paper, we presented a model for computing trust-based reputation for communities of strangers. The model uses the concept of knots, which are sets of members having high levels of trust in each other. We described algorithms for constructing and updating knots and presented a preliminary experimental evaluation which demonstrates the benefits of our knot-aware model.

In very large decentralized communities, the computation of reputation by aggregating the opinions of all community members may be a difficult task, mainly because data from all parts of the network needs to be collected in a secure and consistent manner. The knot-aware model is very suitable for such communities.

We are currently exploring distributed architectures and algorithms for implementing this model. In future work, we plan to investigate and evaluate different clustering algorithms. We also hope to conduct more comprehensive evaluation of the model by obtaining and using additional data-sets.

References

1. ebay, http://www.ebay.com/.
2. Experts-exchange, http://www.experts-exchange.com/.
3. Grouplens, http://www.grouplens.org/.
4. Alfarez Abdul-Rahman and Stephen Hailes. Supporting trust in virtual communities. In *HICSS*, 2000.
5. Sudip Chakraborty and Indrajit Ray. Trustbac: integrating trust relationships into the rbac model for access control in open systems. In *SACMAT '06: Proceedings of the eleventh ACM symposium on Access control models and technologies*, pages 49–58, New York, NY, USA, 2006. ACM Press.
6. Jubin Edachery, Arunabha Sen, and Franz-Josef Brandenburg. Graph clustering using distance-k cliques. In *Graph Drawing*, pages 98–106, 1999.
7. R. Guha, Ravi Kumar, Prabhakar Raghaven, and Andrew Tomkins. Propagation of trust and distrust. In *Proceedings of WWW 04*, pages 403–412. ACM, ACM, May 2004.
8. Audun Josang and Roslan Ismail. The beta reputation system. In *15th Bled Electronic Commerce Conference e-Reality: Constructing the e-Economy*, June 2002.
9. Audun Josang, Roslan Ismail, and Colin Boyd. A survey of trust and reputation systems for online service provision.
10. Sepandar Kamvar, Mario Schlosser, and Hector Garcia-Molina. The EigenTrust Algorithm for Reputation Management in P2P Networks. In *Proceedings of WWW2003*. ACM, 2003.
11. M. Kinateder and K. Rothermel. Architecture and Algorithms for a Distributed Reputation System. In P. Nixon and S. Terzis, editors, *Proceedings of the First International Conference on Trust Management*, volume 2692 of *LNCS*, pages 1–16, Crete, Greece, May 2003. Springer-Verlag.
12. Evangelos Kotsovinos, Petros Zerfos, Nischal M. Piratla, and Niall Cameron. Jiminy: A scalable incentive-based architecture for improving rating quality.
13. Sergio Marti and Hector Garcia-Molina. Taxonomy of trust: Categorizing p2p reputation systems. *Computer Networks*, 50(4):472–484, March 2006.
14. L. Mui, M. Mohtashemi, and A. Halberstadt. A computational model of trust and reputation for e-businesses. In *HICSS '02: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 7*, page 188, Washington, DC, USA, 2002. IEEE Computer Society.
15. Lars Rasmusson and Sverker Jansson. Simulated social control for secure internet commerce. In *NSPW '96: Proceedings of the 1996 workshop on New security paradigms*, pages 18–26. ACM, 1996.
16. Jordi Sabater and Carles Sierra. Review on computational trust and reputation models. *Artificial Intelligence Review*, 24(1):33–60, 2005.
17. Huirong Tian, Shihong Zou, Wendong Wang, and Shiduan Cheng. A group based reputation system for p2p networks. In *ATC*, pages 342–351, 2006.
18. B. Yu and M. Singh. An evidential model of distributed reputation management, 2002.
19. Giorgos Zacharia, Alexandros Moukas, and Pattie Maes. Collaborative reputation mechanisms in electronic marketplaces. *In Proceedings of the 32nd International Conference on System Sciences (HICSS'99)*, 8:8026, January 5–8 1999.