Bayesian Network Trust Model in Peer-to-Peer Networks

Yao Wang, Julita Vassileva

Computer Science Department, University of Saskatchewan, Canada
{yaw181, jiv}@cs.usask.ca

Abstract. In this paper, we propose a Bayesian network-based trust model in peer-to-peer networks. Since trust is multi-faceted, even in the same context, peers still need to develop differentiated trust in different aspects of other peers' behaviors. The peer's needs are different in different situations. Depending on the situation, a peer may need to consider its trust in a specific aspect of another peer's capability or in a combination of multiple aspects. Bayesian networks provide a flexible method to represent differentiated trust and combine different aspects of trust.

1 Introduction

In heterogenous open P2P systems, some peers might be buggy and some might be malicious and provide bad services. Since there is no centralized node to serve as an authority to supervise peers' behaviors and punish peers that behave badly, malicious peers have an incentive to harm other peers to get more benefit because they can get away with their bad behaviors. Mechanisms for trust and reputation can be used to help peers distinguish potential benevolent partners from potential malicious partners and thus provide protection for both consumers and providers of services.

Trust and reputation mechanisms have been proposed for large open environments in e-commerce, distributed computing, recommender systems. Agents are often used to manage and reason about trust and reputation on the behalf of users. In this situation, trust is defined as an agent's belief in attributes such as reliability, honesty and competence of the trusted agent. The reputation of an agent defines an expectation about its behavior, which is based on other agents' observations or information about the agent's past behavior within a specific context at a given time. They are used to help agents make decisions on whether a potential partner is trustworthy to interact with. Here the interaction refers to the activity occurring when an agent gets service from another agent.

Some of the literature on trust and reputation treats the two concepts interchangeably or ambiguously. The two concepts are related, but different. Agent A's trust in agent B is the accumulation of evaluations that agent A has of its past interactions with B. It reflects agent A's subjective viewpoint of B's capability. The reputation of agent B, from agent A's perspective, is the collective evaluation based on other agents' evaluations of B. It is an objective measure for agent B's capability, resulting from the evaluations of many other agents.

2 Bayesian Network-Based Trust Model

We will use a peer-to-peer file sharing application as an example to describe our approach, however this approach is general and can be applied to other domains, like web-services, e-commence, recommender systems or peer-to-peer distributed computing. In peer-to-peer file sharing applications, this approach can also be integrated with current file sharing protocols, such as Gnutella.

In file sharing systems in peer-to-peer networks, each peer plays two roles, the role of file provider offering files to other peers and the role of user using files provided by other peers. In order to distinguish the two roles of each peer, in the rest of the paper, when a peer acts as a file provider, we call it file provider; otherwise, we call it simply agent. An agent builds two kinds of trust. One is the trust in file providers' competence in providing services. The other is the trust in another agent's reliability in providing recommendations about file providers. Here the reliability includes two aspects: whether the agent is truthful in telling its information and whether the agent is trustworthy or not. Since agents are heterogeneous, they judge other agent's behavior by different criteria. If their criteria are similar, one agent can trust another agent. If their criteria are different, they cannot trust each other even if both of them tell the truth. We assume all the agents are truthful in telling their evaluations. So we only take care of the situation where agents have different ways of judging issues, which reflects different user types.

2.1 Trust in A File provider's Competence in Providing Files

In a peer-to-peer network, file providers' capabilities are not uniform. For example, some file providers may be connecting through a high-speed network, so they are able to send files to other agents at a fast speed. Some file providers might like music, so they share a lot of music files. Some may be interested in movies and share some movies. Some may be very picky in file quality, so they only keep and share files with high quality. Therefore, the file provider's capability can be presented in various aspects, such as the download speed, file quality and file type (see Figure 1). The agent's needs are also different in different situations. Sometimes, it might want to know the file provider's overall capability. Sometimes it might only be interested in the file provider's capability in some particular aspect. For instance, an agent wants to download a music file from a file provider. At this time, knowing the file provider's capability in providing music files is more valuable for the agent than knowing the file provider's capability in other aspects. Agents also need to develop differentiated trust in file providers' capabilities. For example, the agent who wants to download a music file from the file provider cares about whether the file provider is able to provide the music file with good quality at a fast speed, which involves the file provider's capabilities in two aspects, quality and speed. How does the agent combine its two trust representations, the trust in the file provider's capability in providing music files with good quality and the trust in the file provider's capability in providing a fast download speed, in order to decide if the file provider is trustworthy?

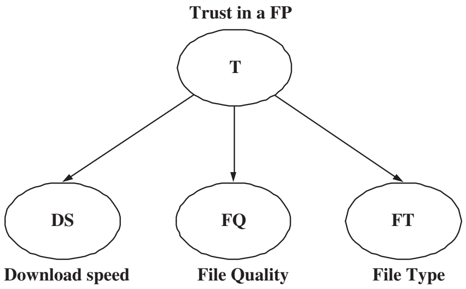

Fig. 1. A bayesian network model

A Bayesian network provides a flexible method to solve the problem. A Bayesian network is a relationship network that uses statistic methods to represent probability relationships between different elements. Its theoretical foundation is the Bayes rule [1].

$$p(h|e) = \frac{p(e|h).p(h)}{p(e)} \tag{1}$$

$p(h)$is the prior probability of hypothesis h; $p(e)$ is the prior probability of evidence e; $p(h|e)$ is the probability of h given e; $p(e|h)$ is the probability of e given h.

A naive Bayesian network is a simple Bayesian network. It is composed of a root node and several leaf nodes. We will use a naive Bayesian network to represent the trust between an agent and a file provider.

Every agent develops a naive Bayesian network for each file provider that it has interacted with. Each Bayesian network has a root node T, which has two values, "satisfying" and "unsatisfying", denoted by 1 and 0, respectively. $p(T = 1)$ represents the value of agent's overall trust in the file provider's competence in providing files. It is the percentage of interactions that are satisfying and measured by the number of satisfying interactions m divided by the total number of interactions n. $p(T = 0)$ is the percentage of not satisfying interactions.

$$p(T = 1) = \frac{m}{n}, \quad \texttt{where } p(T = 1) + p(T = 0) = 1 \tag{2}$$

The leaf nodes under the root node represent the file provider's capability in different aspects. Each leaf node is associated with a conditional probability table (CPT). The node, denoted by FT, represents the set of file types. Suppose it includes five values, *"Music", "Movie", "Document", "Image" and "Software"*. Its CPT is showed in Table 1. It includes two columns of values. Each column follows one constraint, which corresponds to one value of the root node. The sum of values of each column is equal to 1.

$p(FT = "Music"|T = 1)$ is the conditional probability with the condition that an interaction is satisfying. It measures the probability that the file involved

Table 1. The CPT of node FT

	$T = 1$	$T = 0$
Music	$p = (FT = "Music"\|T = 1)$	$p = (FT = "Music"\|T = 0)$
Movie	$p = (FT = "Movie"\|T = 1)$	$p = (FT = "Movie"\|T = 0)$
Document	$p = (FT = "Document"\|T = 1)$	$p = (FT = "Document"\|T = 0)$
Image	$p = (FT = "Image"\|T = 1)$	$p = (FT = "Image"\|T = 0)$
Software	$p = (FT = "Software"\|T = 1)$	$p = (FT = "Software"\|T = 0)$

in an interaction is a music file, given the interaction is satisfying. It can be computed according to the following formula:

$$p(FT = "Music"|T = 1) = \frac{p(FT = "Music", T = 1)}{p(T = 1)} \tag{3}$$

$p(FT = "Music", T = 1)$ is the probability that interactions are satisfying and files involved are music files.

$$p(FT = "Music", T = 1) = \frac{m1}{n} \tag{4}$$

$m1$ is the number of satisfying interactions when files involved are music files.

$p(FT = "Music"|T = 0)$ denotes the probability that files are music files, given interactions are not satisfying. The probabilities for other file types in Table 1 are computed in a similar way.

Node DS denotes the set of download speeds. It has three items,*"Fast", "Medium" and "Slow"*, each of which covers a range of download speed.

Node FQ denotes the set of file qualities. It also has three items, *"High", "Medium" and "Low"*. Its CPT is similar to the one in Table 1.

Here we only take three aspects of trust into account. More relevant aspects can be added in the Bayesian network later to account for user preferences with respect to service.

Once getting nodes' CPTs in a Bayesian network, an agent can compute the probabilities that the corresponding file provider is trustworthy in different aspects by using Bayes rules, such as $p(T = 1|FT = "Music")$ - the probability that the file provider is trustworthy in providing music files, $p(T = 1|FQ = "High")$ - the probability that the file provider is trustworthy in providing files with high quality, $p(T = 1|FT = "Music", FQ = "High")$ - the probability that the file provider is trustworthy in providing music files with high quality. Agents can set various conditions according to their needs. Each probability represents trust in an aspect of the file provider's competence. With the Bayesian networks, agents can infer trust in the various aspects that they need from the corresponding probabilities. That will save agents much effort in building each trust separately, or developing new trust when conditions change.

2.2 Evaluation of an Interaction

A search request in file sharing peer-to-peer applications usually results in a long list of providers for an identical file. An agent can arrange the list according to its trust in these file providers. Then the agent chooses the most trusted file providers in the top of the list to interact with. Agents update their corresponding Bayesian networks after each interaction. If an interaction is satisfying, m and n are both increased by 1 in Formula 4. If it is not satisfying, only n is increased by 1. Two main factors are considered when agents judge an interaction, the degree of their satisfaction with the download speed s_{ds} and the degree of their satisfaction with the quality of downloaded file s_{fq}. The overall degree of agents' satisfaction with an interaction is computed as the following:

$$s = w_{ds} * s_{ds} + w_{fq} * s_{fq}, \quad \texttt{where } w_{ds} + w_{fq} = 1 \tag{5}$$

w_{ds} and w_{fq} denote weights, which indicate the importance of download speed and the importance of file quality to a particular agent (depending on the user's preferences). Each agent has a satisfaction threshold s_t. If $s < s_t$, the interaction is unsatisfying; otherwise, it is satisfying.

2.3 Handing Other Agents' Recommendations

If an agent is not sure about the trustworthiness of a file provider, it can ask other agents to make recommendations for the file provider. The agent can send various recommendation requests according to its needs. For example, if the agent is going to download a movie, it may care about the movie's quality. Another agent may care about the speed. So the request can be "Does the file provider provide good quality movies?". If the agent cares both about the quality and the download speed, the request will be something like "Does the file provider provide good quality files at a fast download speed? ". When other agents receive these requests, they will check their trust-representations, i.e. their Bayesian networks, to see if they can answer such questions. If an agent has downloaded movies from the file provider before, it will send recommendation that contains the value $p(T = 1|FT = "Music", FQ = "High")$ to answer the first request or the value $p(T = 1|FT = "Music", FQ = "High", DS = "Fast")$ to answer the second request. The agent might receive several such recommendations at the same time, which may come from the trustworthy acquaintances, untrustworthy acquaintances, or strangers. If the references are untrustworthy, the agent can discard their recommendations immediately. Then the agent needs to combine the recommendations from trustworthy references and from unknown references together to get the total recommendation for the file provider:

$$r_{ij} = w_t * \frac{\sum_{l=1}^{k} tr_{il} * t_{lj}}{\sum_{l=1}^{k} tr_{il}} + w_s * \frac{\sum_{z=1}^{g} t_{zj}}{g}, \quad \texttt{where } w_t + w_s = 1 \tag{6}$$

r_{ij} is the total recommendation value for the j^{th} file provider that the i^{th} agent gets. k and g are the number of trustworthy references and the number of unknown references, respectively. tr_{il} is the trust that the i^{th} user has in the l^{th} trustworthy reference. t_{lj} is the trust that the l^{th} trustworthy reference has in j^{th} file provider. t_{zj} is the trust that the z^{th} unknown reference has in j^{th} file provider. w_t and w_s are the weights to indicate how the user values the importance of the recommendation from trustworthy references and from unknown references. Since agents often have different preferences and points of view, the agent's trustworthy acquaintances are those agents that share similar preferences and viewpoints with the agent most of time. The agent should weight the recommendations from its trustworthy acquaintances higher than those recommendations from strangers. Given a threshold θ, if the total recommendation value is greater than θ, the agent will interact with the file provider; otherwise, not.

If the agent interacts with the file provider, it will not only update its trust in the file provider, i.e. its corresponding Bayesian network, but also update its trust in the agents that provide recommendations by the following reinforcement learning formula:

$$tr_{ij}^{t+1} = \alpha * tr_{ij}^{t} + (1 - \alpha) * e_{\alpha} \tag{7}$$

tr_{ij}^{t} denotes the trust value that the i^{th} agent has in the j^{th} reference after t recommendations from the j^{th} reference; α is the learning rate - a real number in the interval [0,1]. e_{α} is the new evidence value, which can be -1 or 1. If the value of recommendation is greater than θ and the interaction with the file provider afterwards is satisfying, e_{α} is equal to 1; in the other case, since there is a mismatch between the recommendation and the actual experience with the file provider, the evidence is negative, so e_{α} is -1.

Another way to find if an agent is trustworthy or not in telling the truth is the comparison between two agents' Bayesian networks relevant to an identical file provider. When agents are idle, they can "gossip" with each other periodically, exchange and compare their Bayesian networks. This can help them find other agents who share similar preferences more accurately and faster. After each comparison, the agents will update their trusts in each other according the formula:

$$tr_{ij}^{t+1} = \beta * tr_{ij}^{t} + (1 - \beta) * e_{\beta} \tag{8}$$

The result of the comparison e_{β} is a number in the interval [-1, 1]. β is the learning rate - a real number in the interval [0,1] which follows the constraint $\beta > \alpha$. This is because the Bayesian network collectively reflects an agent's preferences and viewpoints based on all its past interactions with a specific file provider. Comparing the two agents' Bayesian networks is tantamount to comparing all the past interactions of the two agents. The evidence e_{α} in Formula 7 is only based on one interaction. The evidence e_{β} should affect the agent's trust in another agent more than e_{α}.

How do the agents compare their Bayesian networks and how is e_{β} computed? First, we assume all agents have the same structure of Bayesian networks.

We only compare the values in their Bayesian networks. Suppose agent 1 will compare its Bayesian network (see Figure 1) with the corresponding Bayesian network of agent 2. Agent 1 obtains the degree of similarity between the two Bayesian networks by computing the similarity of each pair of nodes (T, DS, FQ and FT), according to the similarity measure based on Clark's distance [2], and then combining the similarity results of each pair of nodes together.

$$e_\beta = 1 - 2 * \sum_{i=1}^{4} (w1_i * c_i), \quad \texttt{where } w1_1 + w1_2 + w1_3 + w1_4 = 1 \tag{9}$$

$$c_1 = \sqrt{\frac{(v1_{11} - v2_{11})^2}{(v1_{11} + v2_{11})^2} + \frac{(v1_{12} - v2_{12})^2}{(v1_{12} + v2_{12})^2}} \tag{10}$$

$$c_i = \frac{\sum_{j=1}^{2} \sqrt{\sum_{l=1}^{h_i} \frac{(v1_{ijl} - v2_{ijl})^2}{(v1_{ijl} + v2_{ijl})^2}}}{2}, \quad \texttt{where } i = 2, 3, 4 \tag{11}$$

$w1_1, w1_2, w1_3$ and $w1_4$ are the weights of the node T, DS, FQ, and FT, respectively, related to agent 1, which indicate the importance of these nodes in comparing two Bayesian networks. c_1, c_2, c_3 and c_4 are the results of comparing agent 1 and agent 2's CPTs about node T, DS, FQ and FT. Since the node T is the root node and it has only one column in its CPT, while other nodes (DS, FQ, FT) are the leaf nodes and have two columns of values in theirs CPTs, we compute c_1 differently from c_2, c_3, and c_4. h_i denotes the number of values in the corresponding node. $h_1 = 3$; $h_2 = 3$; $h_3 = 5$. $v1_{11}$ and $v1_{12}$ are the values of $p(T = 1)$ and $p(T = 0)$ related to agent 1. $v2_{11}$ and $v2_{12}$ are the values of $p(T = 1)$ and $p(T = 0)$ related to agent 2. $v1_{ijl}$ and $v2_{ijl}$ are the values in agent 1's CPTs and agent 2's CPTs, respectively.

The idea of this metric is that agents compute not only their trust values, their CPTs, but also take into account their preferences (encoded as the weights,$w1_1$, $w1_2$, $w1_3$, $w1_4$). So agents with similar preferences, such as the importance of file type, quality, download speed, will weight each other's opinions higher.

3 Evaluation

We evaluate our approach in a simulation of a file sharing system in a peer-to-peer network developed on the JADE 2.5.

3.1 Experimental Setup

For the sake of simplicity, each node in our system plays only one role at a time, either the role of a file provider or the role of an agent. Every agent only knows other agents directly connected with it and a few file providers at the beginning.

Every agent has an interest vector. The interest vector is composed of five elements: *music, movie, image, document and software*. The value of each element indicates the strength of the agent's interests in the corresponding file type. The files the agent wants to download are generated based on its interest vector. Every agent keeps two lists. One is the agent list that records all the other agents that the agent has interacted with and its trust values in these agents. The other is the file provider list that records the known file providers and the corresponding Bayesian networks representing the agent's trusts in these file providers. Each file provider has a capability vector showing its capabilities in different aspects, i.e. providing files with different types, qualities and download speeds.

Our experiments involve 10 different file providers and 40 agents. Each agent will gossip with other agents periodically to exchange their Bayesian networks. The period is 5, which means after each 5 interactions with other agents, the agent will gossip once. $w_{ds} = w_{fq} = 0.5; \alpha = 0.3; \beta = 0.5; w1_1 = w1_2 = w1_3 = w1_4 = 0.25$. The total number of interactions is 1000.

3.2 Results

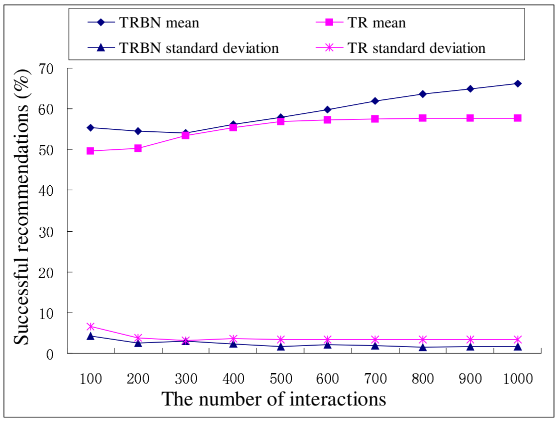

Fig. 2. Trust and reputation system with BN vs. without BN

The goal of the first experiment is to see if a Bayesian network-based trust model helps agents to select file providers that match better their preferences. Therefore we compare the performance (in terms of percentage of successful recommendations, which is the number of successful recommendations over the number of all positive recommendations) of a system consisting of agents with Bayesian network-based trust models and a system consisting of agents (without Bayesian networks, BN) that represent general trust, not differentiated to different aspects. Successful recommendations are those positive recommendations (obtained based on Formula 6) when agents are satisfied with interactions with recommended file providers. If an agent gets a negative recommendation for a file provider, it will not interact with the file provider. We have two configurations in this experiment:

•Trust and reputation system with BN (*TRBN*): the system consists of agents with Bayesian networks-based trust models that exchange recommendations with each other;

•Trust and reputation system without BN (*TR*): the system consists of agents that exchange recommendations, but don't model differentiated trust in file providers.

Figure 2 presents the means and standard deviations of percentages of successful recommendations in the two systems for 10 runs. It shows that the system using Bayesian networks performs slightly better than the system with general trust both in means and in standard deviations. The reason that the curves drop down in the beginning is that the values in the beginning of curves are greatly affected by the initial topology, the influence of which reduces quickly with the increase of interactions. So the latter part of the curves that go up counts more.

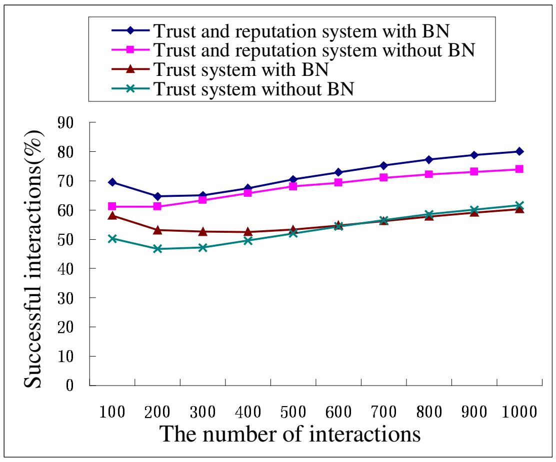

Fig. 3. The Comparison of Four Systems

The goal of the second experiment is to see if the exchange of recommendation values helps agents to achieve better performance (defined as the percentage of successful interactions with file provider, which is the number of successful interactions verse the total number of interactions). This is in fact a measure of confidence in peers' forecast, since peers interact with a file provider only if their trust exceeds a threshold or they get a positive recommendation. For the reason, we compare four configurations:

•Trust and reputation system with BN;

•Trust and reputation system without BN;
•Trust system with BN: the system consists of agents with Bayesian networks-based trust models, which don't exchange recommendations with each other;
•Trust system without BN: the system consists of agents that have no differentiated trust models and don't exchange recommendations with each other.

Figure 3 shows that the two systems, where agents share information with each other, outperform the systems, where agents do not share information. The trust system using Bayesian networks is slightly better than the trust system without using Bayesian networks. There is an anomaly in the case when agents do not share recommendations, since in the end of the curve the system without BN perform better than the system with BN. This could be explained with an imprecise BN due to insufficient experience.

In some sense, an agent's Bayesian network can be viewed as the model of a specified file provider from the agent's personal perspective. In our experiments, we use a very simple naive Bayesian network, which can not represent complex relationships. In the real file-sharing system, the model of file providers might be more complex and required the use of a more complex Bayesian network. Our Bayesian network only involves three factors.

4 Discussion and Related work

How many Bayesian networks can an agent afford to maintain to represent its trust in other agents in the networks? It depends on the size of the network and the likelihood that agents have repeated interactions. Resnick [3] empirically shows that 89.0% of all seller-buyer pairs in eBay conducted just one transaction during a five-month period and 98.9% conducted no more than four. The interactions between the same seller and the same buyer are not repeatable. The buyer's trust in a seller is only based on one direct interaction. The seller's reputation is mostly built on the buyers' having a single experience with the seller. This situation often happens in a very large network or in large e-commence sites. Since there are a large number of sellers and buyers, the chance that a buyer meets the same seller is rare. But if the kind of goods being transacted is only interesting to a small group of people, for example, collectors of ancient coins, the interactions about this kind of goods happen almost exclusively in a small group. So the probability that sellers and buyers have repeated interactions will be high, and they will be able to build trust in each other by our method.

Our approach is useful in situations where two agents can repeatedly interact with each other. In a small-size network, there is no doubt that our approach is applicable. For a large network, our approach is still suitable under the condition that the small-world phenomenon happens. The small-world phenomenon was first discovered in the 1960ies by social scientists. Milgram's experiment showed that people in the U.S. are connected by a short (average length of 6) chain of intermediate acquaintances. Other studies have shown that people tend to interact with other people in their small world more frequently than with people outside. The phenomenon also happens in peer-to-peer networks. Jovanovic [4] shows that the small-world phenomenon occurs in Gnutella. It means that agents are inclined to get files from other agents from a small sub-community. This small sub-community often consists of agents that have similar preferences and viewpoints.

Our approach also requires that agents have compatible ontologies, i.e. their "understanding" of the elements of the Bayesian networks have to be identical or similar at comparision. Although this is a strong assumption, it is feasible if peers are designed by the same person or group or they have similar perferences.

Abdul-Rahman and Hailes [5] capture the most important characteristics of trust and reputation and propose the general structure for developing trust and reputation in a distributed system. Most of the later works in the area follow their ideas, but in different application domain, such as [6], [7], [2], Sabater and Sierra's work [8] extends the notion of trust and reputation into social and ontological dimensions. Social dimension means that the reputation of the group that an individual belongs to also influences the reputation of the individual. Ontological dimension means that the reputation of an agent is compositional. The overall reputation is obtained as a result of the combination of the agent's reputation in each aspect. Our approach integrates these two previous works [5], [8], and applies them to file sharing system in peer-to-peer networks. Another difference between our work and Sabater and Sierra's work is that we use Bayesian networks to represent the differentiated trust at different aspects, other than the structure of ontology. Another difference is that we don't treat the differentiated trusts as compositional. Usually the relationship between different aspects of an agent is not just compositional, but complex and correlative. Our approach provides an easy way to present a complex and correlative relationship. Our approach is also flexible in inferring the trust of an agent for different needs. For example, sometimes we care about the overall trust. Sometimes we only need to know the trust in some specific aspect. This bears parallel with work on distributed user modeling and purpose-based user modeling [9], [10].

Yu and Singh [11], [12] focus on how to use social networks to gather information to compute agents' reputation. They do not deal with differentiated trust and reputation. Cornelli's work [7] is on the area of file sharing in peer-to-peer networks. However, it concentrates on how to prevent the attacks to the reputation system and does not discuss how agents model and compute trust and reputation.

5 Conclusions

In this paper, we propose a Bayesian network-based trust model. Bayesian networks provide a flexible method to represent differentiated trust in different aspects of each other's capability and combine different aspects of trust. We evaluated our approach in a simulation of a file sharing system in a peer-to-peer network. Our experiments show that the system where agents communicate their experiences (recommendations) outperforms the system where agents do not communicate with each other, and that a differentiated trust adds to the performance.

Future work includes adding more aspects in the Bayesian networks, trying to find the key parameters that influence the system performance, and testing the system under other performance measures, for example, how fast an agent can locate a trustworthy service provider. Applying this approach to peer-to-peer systems for computational services is particular promising.

References

1. Heckerman, D.: A tutorial on learning with bayesian networks. Technical report, Microsoft (1995)
2. Montaner, M., López, B., de la Rosa, J.L.: Opinion-based filtering through trust. In: Cooperative Information Agents VI — Proceedings of the 6th International Workshop, CIA 2002. (2002) 164–178
3. Resnick, P., Zeckhauser, R.: Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system. (In: NBER Workshop on Empirical Studies of Electronic Commerce)
4. Jovanovic, M.A.: Modeling large-scale peer-to-peer networks and a case study of gnutella. Master's thesis, University of Cinicnnati (2001)
5. Abdul-Rahman, A., Hailes, S.: Supporting trust in virtual communities. In: Proceedings of the Hawai'i International Conference on System Sciences, Maui, Hawaii (2000)
6. Azzedin, F., Maheswaran, M.: Evolving and managing trust in grid computing systems. In: IEEE Canadian Conference on Electrical & Computer Engineering. (2002)
7. Cornelli, F., Damiani, E., di Vimercati, S.D.C., Paraboschi, S., Samarati, P.: Implementing a reputation-aware gnutella servent. Lecture Notes in Computer Science **2376** (2002) 321–??
8. Sabater, J., Sierra, C.: Regret: a reputation model for gregarious societies. (In: 4^{th}Workshop on Deception, Fraud and Trust in Agent Societies)
9. Niu, X., McCalla, G., Vassileva, J.: Purpose-based user modelling in a multi-agent portfolio management system. In: Proceedings of User Modeling UM03, Johnstown, PA (2003)
10. Vassileva, J., McCalla, G., Greer, J.: Multi-agent multi-user modeling. User Modeling and User-Adapted Interaction (2003)
11. Yu, B., Singh, M.P.: A social mechanism of reputation management in electronic communities. In: Proceedings of Fourth International Workshop on Cooperative Information Agents. (2000) 154–165 36% acceptance rate.
12. Yu, B., Singh, M.P.: Searching social networks. In: Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'03), (ACM Press) 65–72