Application of Belief Propagation to Trust and Reputation Management

Erman Ayday
School of Electrical and Comp. Eng.
Georgia Institute of Technology
Atlanta, GA 30332, USA
Email: eayday@gatech.edu

Faramarz Fekri
School of Electrical and Comp. Eng.
Georgia Institute of Technology
Atlanta, GA 30332, USA
Email: fekri@ece.gatech.edu

***Abstract*—This paper introduces the first application of Belief Propagation (BP) in reputation systems. We view the reputation management as an inference problem, and hence, describe the reputation management problem as computing marginal likelihood distributions from complicated global functions of many variables. However, we observe that computing the marginal probability functions of the reputation variables is computationally prohibitive for large scale reputation systems. Therefore, we propose to utilize the BP algorithm to efficiently (i.e., in linear complexity) compute these marginal probability distributions; leading to a fully iterative probabilistic and BP-based approach (referred to as BP-ITRM). BP-ITRM describes the reputation system on a factor graph, using which we can obtain a qualitative representation of how the service providers (sellers) and consumers (buyers) are related. Further, by using such a graph representation, we compute the marginal probability distribution functions of the variables representing the global reputation values via an iterative message passing algorithm. We show that BP-ITRM significantly outperforms the well-known and commonly used reputation management schemes such as the Averaging Scheme, Bayesian Approach and Cluster Filtering in the presence of attackers. Further, its complexity is linear in the number of service providers and consumers, far exceeding the efficiency of other schemes.**

I. Introduction

Trust and Reputation are crucial requirements for most environments wherein entities participate in various transactions and protocols among each other. The recipient of the service often has insufficient information about the service quality of the service provider before the transaction. Hence, the service recipient should take a prior risk before receiving the actual service. This risk puts the recipient into an unprotected position since he has no opportunity to try the service before he receives it. This problem gives rise to the use of reputation systems in which reputations are determined by rules that evaluate the evidence generated by the past behavior of an entity within a protocol. Hence, after each transaction, a party who receives the service (referred to as the rater) provides (to the central authority) its report about the quality of the service provided for that transaction. The central authority collects the reports and updates the reputations of the service providers. Reputation systems have found widespread adoption in online communities, web services, ad-hoc networks, P2P computing, and e-commerce.

Reputation management systems are subject to various manipulations, launched by the malicious participants. Thus, the success of a reputation scheme depends on the robustness of the mechanism to accurately evaluate the service providers' service qualities (i.e., reputations) and the trustworthiness of the raters based on their reports about the service providers. Further, the emergence of large-scale online services and networks calls for efficient and scalable algorithms to solve for the reputation management problem. Hence, there is a need to develop reliable, scalable and dependable reputation schemes that would also be resilient to various ways a reputation system can be attacked.

In this work, for the first time, we view the reputation management problem as an inference problem and describe the reputation management problem as computing marginal likelihood distributions from complicated global functions of many variables. To solve this problem whose complexity grows exponentially, we resort to use Belief Propagation (BP) [1] whose computational efficiency (i.e., linear in the number of service providers or consumers) is driven by exploring the way in which the global functions factors into a product of simpler local functions. Thus, we introduce the Belief Propagation based Iterative Trust and Reputation Management scheme (BP-ITRM). The work is inspired by earlier work on graph-based iterative probabilistic decoding of low-density parity-check codes [2], the most powerful practically decodable error-control codes known. We believe that the significant benefits offered by the BP algorithms can be tapped in to benefit the field of reputation systems.

In BP-ITRM, the sellers (i.e., service providers) and buyers (i.e., consumers or raters) are represented via a factor graph on which they are arranged as two sets of variable and factor nodes that are connected via some edges. The reputation values of the service providers can be computed by message passing between nodes in the graph. In each iteration of the algorithm, all the variable vertices (service providers) and subsequently all the factor vertices (the raters) pass new messages to their neighbors until convergence. We show that the proposed iterative scheme is reliable (in filtering out malicious/unreliable ratings) while being computationally efficient (i.e., linear in the number of variables). Thus, it can be used as an effective and scalable reputation system in many applications such as online services.

The rest of this paper is organized as follows. In the rest of this section, we summarize the related work. In Section II, we describe the proposed BP-ITRM in detail. Next, in Section III, we evaluate BP-ITRM via computer simulations and compare BP-ITRM with the existing and commonly used reputation management schemes. Further, we analyze BP-ITRM using a mathematical model for the users. Finally, in Section IV, we conclude our paper.

A. Related Work

Several works in the literature have focused on building reputation-management mechanisms [3]. The most famous

and primitive reputation system is the one that is used in eBay. Other well-known web sites such as Amazon, Epinions, and AllExperts use a more advanced reputation mechanism than eBay. Their reputation mechanisms mostly compute the weighted average of the ratings received for a product (or a peer) to evaluate the reputation of a product (or a peer). Hence, these schemes are vulnerable to collaborative attacks by malicious peers. Use of the Bayesian Approach is also proposed in [4]. Finally, [5] proposed to use the *Cluster Filtering* method to distinguish between the reliable and unreliable raters. As we will illustrate via computer simulations, all existing methods are vulnerable to sophisticated attacks such as RepTrap [6] since none of these schemes are designed considering the noise and the incomplete information in the system. Inspired by the earlier work on iterative decoding of error-control codes in the presence of stopping sets, we developed an algebraic iterative algorithm for reputation management [7] and adversary detection [8]. Here, we propose a new scheme based on the key observation that the reputation management problem can be formulated as solving for marginal likelihood functions of many variables by using the BP algorithm. As we will illustrate, comparison with the existing schemes suggests that the proposed BP-ITRM has superior performance (i.e., accuracy, scalability and robustness against attacks).

II. Belief Propagation Based Iterative Trust and Reputation Management

In the reputation management problem, we wish to make statistical inference about the reputations of service providers based on past observations. That is, given the past data evidence, what is the likelihood (probability) of the reputation values being "good" or "bad"? Here, the interpretation of probability is a Bayesian one. We formulate this problem as finding the marginal probability distributions problem. Unfortunately, computing such probability distributions is computationally prohibitive for large-scale systems. However, the proposed algorithm (referred to as BP-ITRM) shows that this problem can be solved, fortunately, by applying the Belief Propagation (BP) algorithm (in linear complexity).

We assume two different sets in the reputation system: i) the set of service providers, $\mathbb{S}$, and ii) the set of service consumers (raters), $\mathbb{U}$. We note that these two sets are not necessarily disjoint. Transactions occur between service providers and consumers, and consumers (raters) provide feedbacks in the form of ratings about service providers after each transaction. As in every reputation management mechanism, we have two main goals: 1. computing the service quality (reputation) of the peers who provide a service (henceforth referred to as Service Providers or SPs) by using the feedbacks from the peers who used the service (referred to as the raters), and 2. determining the trustworthiness of the raters by analyzing their feedback about SPs. Let G_j be the reputation value of the SP j ($j \in \mathbb{S}$) and T_{ij} be the rating that the rater i ($i \in \mathbb{U}$) reports about the SP j ($j \in \mathbb{S}$), whenever a transaction is completed between the two peers. Moreover, let R_i denote the trustworthiness of the peer i ($i \in \mathbb{U}$) as a rater. In other words, R_i represents the amount of confidence that the reputation system has about the correctness of any feedback/rating provided by the rater i. We assume there are u raters and s SPs in the system (i.e., $|\mathbb{U}| = u$ and $|\mathbb{S}| = s$). We let $\mathbb{G} = \{G_j : j \in \mathbb{S}\}$ and $\mathbb{R} = \{R_i : i \in \mathbb{U}\}$ be the collection of variables representing the reputations of the SPs and the trustworthiness values of the raters, respectively. Further, let $\mathbb{T}$ be the $s \times u$ SP-rater matrix that stores the rating values (T_{ij}), and $\mathbb{T}_i$ be the set of ratings provided by the rater i. For simplicity of presentation, we assume that the rating values are from the set $\Upsilon = \{0, 1\}$. The extension in which rating values can take any real number can be developed similarly.

The reputation management problem can be viewed as finding the marginal probability distributions of each variable in $\mathbb{G}$, given the observed data (i.e., evidence). There are s marginal probability functions, $p(G_j|\mathbb{T}, \mathbb{R})$, each of which is associated with a variable G_j; the reputation value of SP j. We formulate the problem by considering the global function $p(\mathbb{G}|\mathbb{T}, \mathbb{R})$, which is the joint probability distribution function of the variables in $\mathbb{G}$ given the rating matrix and the trustworthiness values of the raters. Then, clearly, each marginal probability function $p(G_j|\mathbb{T}, \mathbb{R})$ may be obtained as follows[1]:

$$p(G_j|\mathbb{T}, \mathbb{R}) = \sum_{\mathbb{G}\setminus\{G_j\}} p(\mathbb{G}|\mathbb{T}, \mathbb{R}). \quad (1)$$

Unfortunately, the number of terms in (1) grows exponentially with the number of variables, making it infeasible for large-scale systems. However, we propose to factorize (1) to local functions f_i using a factor graph and utilize the BP algorithm to calculate the marginal probability distributions in linear complexity. A factor graph is a bipartite graph containing two sets of nodes (corresponding to variables and factors) and edges incident between two sets. Following [9], we form a factor graph by setting a variable node for each variable G_j, a factor node for each function f_i, and an edge connecting variable node j to the factor node i if and only if G_j is an argument of f_i.

Therefore, we arrange the collection of the raters and the SPs together with their associated relations (i.e., the ratings of the SPs by the raters) as a bipartite (or factor) graph, as in Fig. 1. In this representation, each rater peer corresponds to a factor node in the graph, shown as a square. Each SP is represented by a variable node shown as a hexagon in the graph. Each report/rating is represented by an edge from the factor node to the variable node. Hence, if a rater i ($i \in \mathbb{U}$) has a report about the SP j ($j \in \mathbb{S}$), we place an edge with value T_{ij} from the factor node i to the variable node representing the SP j.

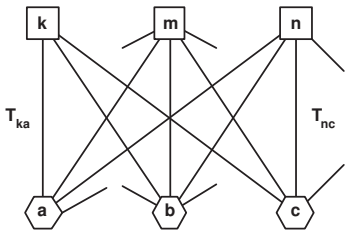

Fig. 1: Factor graph between the SPs and the raters.

Next, we assume that the global function $p(\mathbb{G}|\mathbb{T}, \mathbb{R})$ factors into products of several local functions, each having a subset of variables from $\mathbb{G}$ as arguments as follows:

$$p(\mathbb{G}|\mathbb{T}, \mathbb{R}) = \frac{1}{Z} \prod_{i \in \mathbb{U}} f_i(\mathcal{G}_i, \mathbb{T}_i, R_i), \quad (2)$$

where Z is the normalization constant and $\mathcal{G}_i$ is a subset of $\mathbb{G}$. Hence, in the graph representation of Fig. 1, each factor node is associated with a local function and each local function f_i represents the probability distributions of its arguments

[1]The notation $\mathbb{G}\setminus\{G_j\}$ implies all variables in $\mathbb{G}$ except G_j.

given the trustworthiness value and the existing ratings of the associated rater.

We now introduce the messages between the factor and the variable nodes to compute the marginal distributions using BP. To that end, we choose an arbitrary factor graph as in Fig. 1 and describe message exchanges between rater k and SP a. We represent the set of neighbors of the variable node (SP) a and the factor node (rater) k as $\mathbf{N_a}$ and $\mathbf{N_k}$, respectively[2]. Further, let $\Xi = \mathbf{N_a}\backslash\{k\}$ and $\Delta = \mathbf{N_k}\backslash\{a\}$. The BP algorithm iteratively exchanges the probabilistic messages between the factor and the variable nodes in Fig. 1, updating the degree of beliefs on the reputation values of the SPs as well as the confidence of the raters on their ratings (i.e., trustworthiness values) at each step, until convergence. Let $\mathbb{G}^{(\nu)} = \{G_j^{(\nu)} : j \in \mathbb{S}\}$ be the collection of variables representing the values of the variable nodes at the iteration ν of the algorithm. We denote the messages from the variable nodes to the factor nodes and from the factor nodes to the variable nodes as μ and λ, respectively. The message $\mu_{a\to k}^{(\nu)}(G_a^{(\nu)})$ denotes the probability of $G_a^{(\nu)} = \ell$, $\ell \in \{0,1\}$, at the ν^{th} iteration. On the other hand, $\lambda_{k\to a}^{(\nu)}(G_a^{(\nu)})$ denotes the probability that $G_a^{(\nu)} = \ell$, for $\ell \in \{0,1\}$, at the ν^{th} iteration given T_{ka} and R_k.

The message from the factor node k to the variable node a at the ν^{th} iteration is formed using the principles of the BP as

$$\lambda_{k\to a}^{(\nu)}(G_a^{(\nu)}) = \sum_{\mathbb{G}^{(\nu-1)}\backslash\{G_a^{(\nu-1)}\}} f_k(\mathcal{G}_k, \mathbb{T}_k, R_k^{(\nu-1)}) \prod_{x\in\Delta} \mu_{x\to k}^{(\nu-1)}(G_x^{(\nu-1)}), \tag{3}$$

where $\mathcal{G}_k$ is the set of variable nodes which are arguments of the local function f_k at the factor node k. Further, $R_k^{(\nu-1)}$ (the trustworthiness of rater k calculated at the end of $(\nu-1)^{th}$ iteration) is a value between zero and one and can be calculated as follows:

$$R_k^{(\nu-1)} = 1 - \frac{1}{|N_k|} \sum_{i\in N_k} \sum_{x\in\{0,1\}} |T_{ki} - x| \mu_{i\to k}^{(\nu-1)}(x). \tag{4}$$

The above equation can be interpreted as one minus the average inconsistency of rater k calculated using the messages it received from its neighbors. Using (3), it can be shown that $\lambda_{k\to a}^{(\nu)}(G_a^{(\nu)}) \propto p(G_a^{(\nu)}|T_{ka}, R_k^{(\nu-1)})$, where

$$\begin{aligned} &p(G_a^{(\nu)}|T_{ka}, R_k^{(\nu-1)}) = \\ &\Big[(R_k^{(\nu-1)} + \frac{1-R_k^{(\nu-1)}}{2})T_{ka} + \frac{1-R_k^{(\nu-1)}}{2}(1-T_{ka})\Big]G_a^{(\nu)} + \\ &\Big[\frac{1-R_k^{(\nu-1)}}{2}T_{ka} + (R_k^{(\nu-1)} + \frac{1-R_k^{(\nu-1)}}{2})(1-T_{ka})\Big](1-G_a^{(\nu)}). \end{aligned} \tag{5}$$

This resembles the belief/pleusability concept of the Dempster-Shafer Theory [10]. Given $T_{ka} = 1$, $R_k^{(\nu-1)}$ can be considered as the belief of the k^{th} rater that the $G_a^{(\nu)}$ value is one (at the ν^{th} iteration). In other words, in the eyes of rater k, the $G_a^{(\nu)}$ value is equal to one with probability $R_k^{(\nu-1)}$. Thus, $(1-R_k^{(\nu-1)})$ corresponds to the uncertainty in the belief of rater k. In order to remove this uncertainty and express $p(G_a^{(\nu)}|T_{ka}, R_k^{(\nu-1)})$ as the probabilities that $G_a^{(\nu)}$ is zero and one, we distribute the uncertainty uniformly between two outcomes (one and zero). Hence, in the eyes of the k^{th} rater, $G_a^{(\nu)}$ value is equal to one with probability $(R_k^{(\nu-1)} + (1-R_k^{(\nu-1)})/2)$, and zero with probability $((1-R_k^{(\nu-1)})/2)$. We note that a similar statement holds for the case when $T_{ka} = 0$. The above computation must be performed for every neighbors of each factor nodes. This finishes the first half of the ν^{th} iteration.

During the second half of the ν^{th} iteration, the variable nodes generate their messages (μ) and send to their neighbors. Variable node a forms $\mu_{a\to k}^{(\nu)}(G_a^{(\nu)})$ by multiplying all information it receives from its neighbors excluding the factor node k. Hence, the message from variable node a to the factor node k at the ν^{th} iteration is given by

$$\mu_{a\to k}^{(\nu)}(G_a^{(\nu)}) = \frac{1}{\sum_{h\in\{0,1\}} \prod_{i\in\Xi} \lambda_{i\to a}^{(\nu)}(h)} \times \prod_{i\in\Xi} \lambda_{i\to a}^{(\nu)}(G_a^{(\nu)}) \tag{6}$$

This computation is repeated for every neighbors of each variable node. The algorithm proceeds to the next iteration in the same way as the ν^{th} iteration. We note that the iterative algorithm starts its first iteration by computing $\lambda_{k\to a}^{(1)}(G_a^{(1)})$ in (3). However, instead of calculating in (4), the trustworthiness value R_k from the previous execution of BP-ITRM is used as initial values in (5).

The iterations stop when all variables in $\mathbb{G}$ converge. Therefore, at the end of each iteration, the reputations are calculated for each SP. To calculate the reputation value $G_a^{(\nu)}$, we first compute $\mu_a^{(\nu)}(G_a^{(\nu)})$ using (6) but replacing Ξ with $\mathbf{N_a}$, and then we set $G_a^{(\nu)} = \sum_{i=0}^{1} i\mu_a^{(\nu)}(i)$.

III. Evaluation of BP-ITRM

Here, we wish to evaluate the proposed BP-ITRM using a mathematical model for the users. We consider slotted time throughout this discussion. For each time-slot (or epoch), the iterative reputation algorithm is executed using the input parameters $\mathbb{R}$ and $\mathbb{T}$ to output the reputation values after convergence. We assumed that the rating values are either 0 or 1. We let $\hat{G}_j$ denote the actual value of the reputation of the SP j ($j \in \mathbb{S}$), where $\hat{G}_j \in \{0,1\}$ (1 represents a good service quality). Further, the quality of each service provider remains unchanged during time-slots. Ratings generated by the non-malicious raters are uniformly distributed among the SPs (i.e., their ratings/edges in the graph representation are distributed uniformly among SPs). We also assume the rating r_h (provided by a non-malicious rater) is a random variable with Bernoulli distribution, where $Pr(r_h = \hat{G}_j) = p_c$ and $Pr(r_h \neq \hat{G}_j) = (1-p_c)$. To facilitate future references, frequently used notations are listed in Table I.

$\mathbb{U}_M$: The set of malicious raters
$\mathbb{U}_R$: The set of non-malicious raters
r_h: Report (rating) given by a non-malicious rater
r_m: Report (rating) given by a malicious rater
d: Total number of newly generated ratings, per time-slot, per non-malicious rater
b: Total number of newly generated ratings, per time-slot, per malicious rater

TABLE I: Notations and definitions.

A. Malicious User Model

We consider two major attacks that are common for any reputation management mechanisms:

Bad mouthing: Malicious raters collude and attack the service providers with the highest reputation by giving low ratings

[2]Neighbors of a SP are the set of raters who rated the SP while neighbors of a rater are the SPs whom it rated.

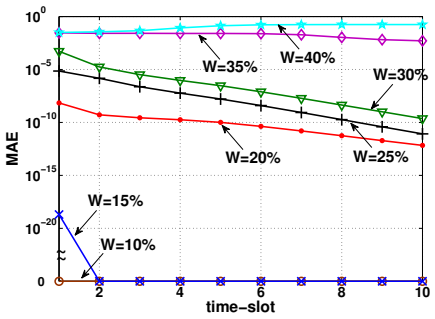

Fig. 2: MAE performance of BP-ITRM versus time when W of the existing raters are malicious in RepTrap [6].

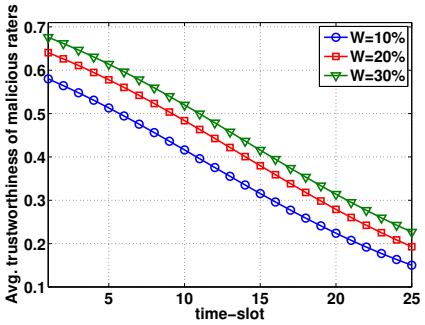

Fig. 3: Average trustworthiness of malicious raters versus time for BP-ITRM when W of the existing raters are malicious in RepTrap [6].

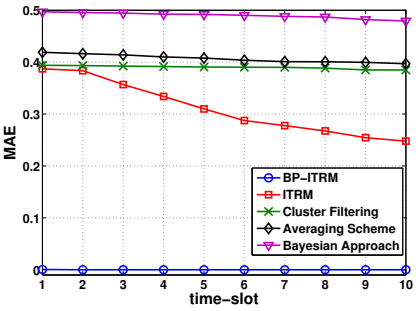

Fig. 4: MAE performance of various schemes when 30% of the existing raters become malicious in RepTrap [6].

in order to undermine them. It is also noted that in addition to the malicious peers, in some applications, bad mouthing may be originated by a group of selfish peers who attempt to weaken high-reputation SPs in the hope of improving their own chances as providers.

Ballot stuffing: Malicious raters collude to increase the reputation values of peers with low reputations. In some applications, this could be mounted by a group of selfish consumers attempting to favor their allies.

We make the following assumptions for modeling the adversary which is also known as the RepTrap attack [6]. RepTrap is believed to be a strong attack against the reputation management system. We assumed that the malicious raters initiate bad mouthing[3]. Further, all the malicious raters collude and attack the same subset Γ of SPs in each time-slot (which represents the strongest attack), by rating those SPs as $r_m = 0$. In other words, we denote by Γ the set of size b in which every victim SP has one edge from each of the malicious raters. The subset Γ is chosen to include those SPs who have the highest reputation values but received the lowest number of ratings from the non-malicious raters (assuming that the attackers have this information). To the advantage of malicious raters, we assumed that a total of T time-slots had passed since the initialization of the system and a fraction of the existing raters change behavior and become malicious after T time-slots. In other words, malicious raters behaved like reliable raters and increased their trustworthiness values before mounting their attacks at the $(T+1)^{th}$ time-slot. We will evaluate the performance for the time-slot $(T+1)$.

B. Simulations

We compared the performance of BP-ITRM with three well-known and commonly used reputation management schemes: 1) *The Averaging Scheme* (which is widely used in eBay or Amazon), 2) *Bayesian Approach* [4], and 3) *Cluster Filtering* [5]. Further, we compared BP-ITRM with our previous method [7] (referred to as ITRM).

We assumed that d (in Table I) is a random variable with Yule-Simon distribution, which resembles the power-law distribution used in modeling online systems, with the probability mass function $f_d(d;\rho) = \rho B(d, \rho+1)$, where B is the Beta function. Further, we set $T = 50$, $b = 5$, $p_c = 0.8$, $\rho = 1$, $|\mathbb{U}| = 100$ and $|\mathbb{S}| = 100$. We assumed the adversary model in Section III-A. In the following, we measured the performance of BP-ITRM, for each time-slot, as the mean absolute error (MAE) $|G_j - \hat{G}_j|$, averaged over all the SPs that are under attack. The time-slots in all the plots are shown after subtracting the offset-time $T = 50$.

First, we evaluated the MAE performance of BP-ITRM for different fractions of malicious raters ($W = \frac{\mathbb{U}_M}{\mathbb{U}_M+\mathbb{U}_R}$), at different time-slots in Fig. 2. We observed that BP-ITRM provides significantly low errors for up to $W = 40\%$ malicious raters. Next, in Fig. 3, we show the change in the average trustworthiness (R_i values) of malicious raters with time. We conclude that the trustworthiness values of the malicious raters decrease over time, and hence, the impact of the malicious ratings vanishes over time. Further, Fig. 4 illustrates the comparison of BP-ITRM with the other well-known and commonly used reputation systems for bad mouthing when the fraction of malicious raters (W) is 30%. It is clear that BP-ITRM outperforms all the other techniques significantly. In all these simulations, the average number of iterations for BP-ITRM is around 10 and it decreases with time and with decreasing fraction of malicious raters.

In most reputation systems, the adversary causes the most serious damage by introducing newcomer raters to the system. Since it is not possible for the system to know the trustworthiness of the newcomer raters, the adversary may introduce newcomer raters to the systems and attack the SPs using those raters. To study the effect of newcomer malicious raters to the scheme, we introduced 100 more raters as newcomers. Hence, we had $|\mathbb{U}| = 200$ raters and $|\mathbb{S}| = 100$ SPs in total. We assumed that the rating values are either 0 or 1, r_h is a random variable with Bernoulli distribution as before, and malicious raters choose SPs from Γ and rate them as $r_m = 0$ (this particular attack scenario does not represent the RepTrap attack). In Fig. 5 we observed that BP-ITRM still provides significantly low MAE and preserves its superiority over the other schemes.

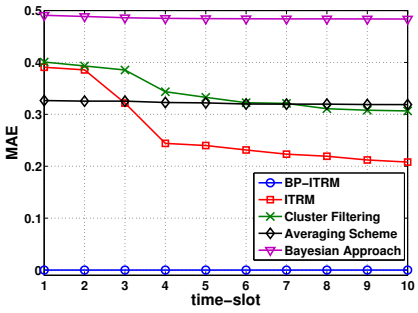

Fig. 5: MAE performance of various schemes when 30% of the newcomer raters are malicious.

From these simulation results, we conclude that BP-ITRM significantly outperforms the Averaging Scheme, Bayesian

[3]It is worth nothing that even though we use the bad-mouthing attack, similar counterpart results hold for ballot stuffing and combinations of bad mouthing and ballot stuffing.

Approach and Cluster Filtering in the presence of attackers. We identify that our non-probabilistic iterative scheme ITRM [7] is the closest in performance to BP-ITRM. This emphasizes the robustness of using iterative algorithms for reputation management. Finally, assuming u raters and s SPs, we obtained the computational complexity of BP-ITRM as $\mathbf{max}(\mathrm{O}(cu), \mathrm{O}(cs))$ in the number of multiplications, where c is a small number representing the average number of rating edges per rater. In contrast, Cluster Filtering suffers quadratic complexity versus number of raters (or SPs).

C. The ϵ-Optimal Scheme

Using the adopted models for various peers, it is natural to ask if BP-ITRM maintain any optimality in any sense. We declare a reputation scheme to be ϵ-optimal if the mean absolute error (MAE) ($|G_j - \hat{G}_j|$) is less than or equal to ϵ for every SP j. Thus, for a fixed ϵ, we wish to obtain the conditions for an ϵ-optimal scheme. It can be shown that, equivalently, we require BP-ITRM to iteratively reduce the impact of malicious raters and decrease the error in the reputation values of the SPs below ϵ until it converges.

Lemma 1: (Sufficient Condition): The error in the reputation values of the SPs decreases with each successive iterations (until convergence) if $G_a^{(2)} > G_a^{(1)}$ is satisfied with high probability for every SP a ($a \in \mathbb{S}$) with $\hat{G}_a = 1$[4].

Proof: Let $G_a^{(\omega)}$ and $G_a^{(\omega+1)}$ be the reputation value of an arbitrary SP a with $\hat{G}_a = 1$ calculated at the $(\omega)^{th}$ and $(\omega+1)^{th}$ iterations, respectively. $G_a^{(\omega+1)} > G_a^{(\omega)}$ if the following is satisfied at the $(\omega+1)^{th}$ iteration.

$$\prod_{j \in \mathbb{U}_R \cap \mathbf{N_a}} \frac{2p_c R_j^{(w+1)} + 1 - R_j^{(w+1)}}{-2p_c R_j^{(w+1)} + 1 + R_j^{(w+1)}} \prod_{j \in \mathbb{U}_M \cap \mathbf{N_a}} \frac{1 - \hat{R}_j^{(w+1)}}{1 + \hat{R}_j^{(w+1)}} > \prod_{j \in \mathbb{U}_R \cap \mathbf{N_a}} \frac{2p_c R_j^{(w)} + 1 - R_j^{(w)}}{-2p_c R_j^{(w)} + 1 + R_j^{(w)}} \prod_{j \in \mathbb{U}_M \cap \mathbf{N_a}} \frac{1 - \hat{R}_j^{(w)}}{1 + \hat{R}_j^{(w)}}, \quad (7)$$

where $R_j^{(w)}$ and $\hat{R}_j^{(w)}$ are the trustworthiness values of a reliable and malicious rater calculated as in (4) at the w^{th} iteration, respectively.

Given $G_a^{(\omega)} > G_a^{(\omega-1)}$ holds at the ω^{th} iteration, we would get $\hat{R}_j^{(w)} > \hat{R}_j^{(w+1)}$ for $j \in \mathbb{U}_M \cap \mathbf{N_a}$ and $R_j^{(w+1)} \geq R_j^{(w)}$ for $j \in \mathbb{U}_R \cap \mathbf{N_a}$. Thus, (7) would hold for the $(w+1)^{th}$ iteration. On the other hand, if $G_a^{(\omega)} < G_a^{(\omega-1)}$, we get $\hat{R}_j^{(w)} < \hat{R}_j^{(w+1)}$ for $j \in \mathbb{U}_M \cap \mathbf{N_a}$ and $R_j^{(w+1)} < R_j^{(w)}$ for $j \in \mathbb{U}_R \cap \mathbf{N_a}$. Hence, (7) is not satisfied at the $(w+1)^{th}$ iteration. Therefore, if $G_a^{(\omega)} > G_a^{(\omega-1)}$ holds for some iteration ω, then the BP-ITRM algorithm reduces the error on the global reputation value (G_a) until the iterations stop, and hence, it is sufficient to satisfy $G_j^{(2)} > G_j^{(1)}$ with high probability for every SP j with $\hat{G}_j = 1$ to guarantee that BP-ITRM iteratively reduces the impact of malicious raters until it stops. ■

Once the sufficient condition is met, the probability for ϵ-optimality can be obtained[5]. Now, we illustrate performance of our scheme in terms of ϵ for which BP-ITRM is an ϵ-optimal scheme based on our analytical results. As before, we assumed that d is a random variable with Yule-Simon distribution (with $\rho = 1$). The parameters we used are $|\mathbb{U}_M| + |\mathbb{U}_R| = 100$, $|\mathbb{S}| = 100$, $\rho = 1$, $T = 50$, $b = 5$ and $p_c = 0.8$. In Fig. 6, we illustrate the average ϵ (ϵ_{av}) for which BP-ITRM is an ϵ-optimal scheme with high probability for different fractions of malicious raters. We observed that BP-ITRM provides significantly small error values for up to 30% malicious raters.

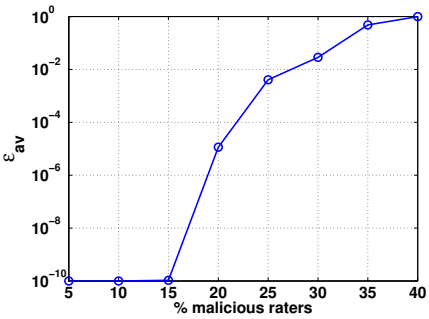

Fig. 6: The average ϵ values for which BP-ITRM is an ϵ-optimal scheme with high probability versus fraction of malicious raters.

IV. Conclusion

In this paper, we introduced the first application of the Belief Propagation algorithm to solve for the inference problem arising in reputation systems. We presented the Belief Propagation based Iterative Trust and Reputation Management Scheme (BP-ITRM). BP-ITRM is a graph-based reputation management system in which service providers and raters are arranged as two sets of variable and factor nodes and the reputation values of service providers are computed by message passing between these nodes in the graph until the convergence. The proposed BP-ITRM is a robust mechanism to evaluate the quality of the service of the service providers from the ratings received from the raters. Moreover, it effectively evaluates the trustworthiness of the raters. We showed that the complexity of the proposed scheme grows only linearly with the number of service providers (or raters) in the system. Further, we studied BP-ITRM in a detailed analysis and showed the robustness using computer simulations. We also compared BP-ITRM with some well-known reputation management schemes and showed the superiority of our scheme both in terms of robustness against various attacks and efficiency.

[4]The opposite must hold for any SP with $\hat{G}_a = 0$.

[5]We omitted the expression due to page limit.

References

[1] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., 1988.

[2] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity check codes under message-passing decoding," *IEEE Transactions on Information Theory*, vol. 47, pp. 599–618, Feb. 2001.

[3] P. Resnick, R. Zeckhauser, E. Friedman, and K. Kuwabara, "Reputation systems: facilitating trust in internet interactions," *Communications of the ACM*, vol. 43, no. 12, pp. 45–48, 2000.

[4] S. Buchegger and J.Boudec, "Coping with false accusations in misbehavior reputation systems for mobile ad-hoc networks," *EPFL-DI-ICA Technical Report IC/2003/31*, 2003.

[5] C. Dellarocas, "Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior," *EC '00: Proceedings of the 2nd ACM conference on Electronic commerce*, pp. 150–157, 2000.

[6] Y. Yang, Q. Feng, Y. L. Sun, and Y. Dai, "RepTrap: a novel attack on feedback-based reputation systems," *SecureComm '08: Proceedings of the 4th International Conference on Security and Privacy in Communication Networks*, pp. 1–11, 2008.

[7] E. Ayday, H. Lee, and F. Fekri, "An iterative algorithm for trust and reputation management," *ISIT '09: Proceedings of IEEE International Symposium on Information Theory*, 2009.

[8] ——, "Trust management and adversary detection in delay tolerant networks," *In Proceedings of IEEE Military Communications Conference (MILCOM)*, 2010.

[9] F. Kschischang, B. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, pp. 498–519, Feb. 2001.

[10] G. Shafer, *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, N.J., 1976.