Trustworthiness and Untrustworthiness Inference with Group Assignment

Xinxin Fan[1(✉)], Danyang He[1,2], and Jingping Bi[1]

[1] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
{fanxinxin,hedanyang,bjp}@ict.ac.cn
[2] Dalian University of Technology, Dalian, Liaoning, China

Abstract. Diverse strategically misbehaved entities have severely degraded the core-functions of trust-enabled interactional networks. At present, it is still a hard problem to identify them owing to the complexities of malicious behaviors, such as on-off attack, colluding attack, etc. In this paper, we propose a belief propagation-based algorithm MapTrust to quantitatively and qualitatively infer entity's trustworthiness and untrustworthiness. Three primary contributions are included: (i) we define removal probability for each pair of interacted entities via pairwise feedback-ratings; (ii) we propose a novel cross-iteration fashion to infer trustworthiness and untrustworthiness values. The cross-iteration fashion not only declines time overhead compared to sequential iteration method, but it also supports a convenient manipulation, i.e. we can flexibly initiate group affinity; (iii) we launch extensive experiments using synthetic and real-world datasets to verify the efficiency of our proposed MapTrust. The experimental results show our proposed MapTrust dramatically outperforms Monte Carlo Markove Chain and Random algorithms against four representative attacks.

Keywords: Trustworthiness propagation · Group assignment
Belief propagation · Trust-enabled interactional networks

1 Introduction

Interactional networks enable massive global-scattered distributed resources over the Internet to benefit customers through providing services on-demand, such as cloud platforms, P2P networks, WSNs, online social networks and eCommerce, IoTs, etc. Nevertheless, the emergence of various misbehaved entities, especially the strategically malicious collectives which can occasionally behave honestly alike good entities, severely damages the networking utilities and degrades the service capability. Accordingly, the reputation-based trust management, as an effective security mechanism, has been proposed and successfully applied in the real-world systems, such as Amazon, Alibaba, etc. Usually, the trust metrics compute a unique global trust score for each entity to denote how trustworthy it is through aggregating both direct and recommended feedback-ratings. The higher

© Springer International Publishing AG, part of Springer Nature 2018
H. Jin et al. (Eds.): ICWS 2018, LNCS 10966, pp. 389–404, 2018.
https://doi.org/10.1007/978-3-319-94289-6_25

the global trust score, the more reliable the entity. For misbehaved entities, they attempt to get high global trust via strategically collusive manipulations to subvert the system, and some malicious entities can succeed indeed. Our previous work [4] has verified the camouflage and spy entities can indeed gain high global trust scores using the popular trust model-EigenTrust [9]. As analyzed in [4], the commonly used uniformly distributed trust propagation kernel almost fails to block strategically misbehaved entities from gaining trust propagation. It will inversely yield high global trust scores for the strategically misbehaved entities, i.e. camouflage and spy entities. Keeping this weakness in mind, we try to address this sophisticated security question from another perspective-trustworthiness and untrustworthiness computation, i.e. on the one hand, we admit the existence of honest behaviors of these strategically misbehaved entities; on the other hand, we portend their potentially co-existing trustworthiness and untrustworthiness values.

In this paper, we propose an effective behavioral trustworthiness and untrustworthiness inference algorithm MapTrust through calculating a fine-grained marginal probability using belief propagation (BP) algorithm, replacing the traditional global trust aggregation fashion [4,9,16]. In MapTrust, we define belief/unbelief propagating kernel for each pair of connected/interacted entities. In this way, even though a strategically misbehaved entity can get high feedback-ratings, it cannot be recognized as trustworthy owning to the low removal probability with pre-trusted entities which are initially assigned trustworthy entities by the networked system [4,9,16]. For a good entity recently joined into the network, although it cannot get many feedback-ratings, it still may be recognized as trustworthy due to high removal probability with pre-trusted entities. This breaks through the deficiency of referring single global trust as interactional criterion. Our main contributions can conclude as follows.

(i) We define pairwise removal probability for each pair of interacted entities using the gravitation model. The removal probability, as the basis of marginal probability aggregation, stands for the affinity for a pair of entities.
(ii) We propose a novel cross-iteration fashion to compute marginal probability through two-layer "message-passing". This cross-iteration fashion also takes two facets of advantages. Firstly, it declines the time cost compared to sequential iteration. Secondly, it also brings a flexible setting on intra/inter-group affinity.
(iii) We launch extensive experiments using both synthetic and real-world datasets to investigate the efficiency of our proposed MapTrust, and compared with Monte Carlo Markove Chain (MCMC) and Random algorithms. The experimental results show that our MapTrust not only assigns different categories of entities into adequate groups, but it can also appropriately calculate fine-grained trustworthiness and untrustworthiness values for each entity.

The rest of this paper are organized as follows. Section 2 introduces the radical components in trust-enabled interactional networks. Section 3 details MapTrust

from the perspectives of removal probability setting and trustworthiness and untrustworthiness inference. We conduct extensive experiments to evaluate the efficiency of our MapTrust in Sect. 4, present related work in Sect. 5 and conclude the paper in Sect. 6.

2 Radical Components in Trust-Enabled Interactional Networks

2.1 Local Feedback-Rating Aggregate

In a trust-enabled interactional network, the service consumer will give the service provider (seller) a feedback-rating to state his/her opinion on the quality of service. If satisfied, the service consumer would give the service provider a positive feedback-rating; otherwise a negative feedback-rating if unsatisfied. Let ς_{ij} denote the number of satisfied transactions between entities i and j, and τ_{ij} denote the number of unsatisfied transactions. To date, the local feedback-rating aggregate primarily contains two manners. One is the positive feedback-rating ratio [4,5,9,16], it can be straightly defined as:

$$s_{ij} = \begin{cases} \frac{\varsigma_{ij}}{\varsigma_{ij}+\tau_{ij}+1} \frac{\tau_{ij}}{\varsigma_{ij}+\tau_{ij}+1} \leq \theta \\ \frac{1}{2} \quad otherwise \end{cases}, \tag{1}$$

where constant θ implies the good entities can be allowed to misbehave in a subtle probability (usually 0.5) owing to some unintentional reasons.

The other manner is to utilize beta-function [7,8]:

$$f(p|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1}, \tag{2}$$

where Γ is gamma function, $0 \leq p \leq 1, \alpha, \beta > 0, p \neq 0$ if $\alpha < 1$ and $p \neq 1 if \beta < 1$. The local feedback-rating value is defined as the probability expectation $E(p)$:

$$\frac{\alpha}{\alpha+\beta} = \frac{\varsigma_{ij}+1}{\varsigma_{ij}+\tau_{ij}+2}, \tag{3}$$

where $\alpha = \varsigma_{ij} + 1$, $\beta = \tau_{ij} + 1$. Obviously, the two manners reflect similar meanings, i.e. both Formulas (1) and (3) aim at mirroring the potential local trustworthiness to some extent. In this paper, we choose the former to aggregate local feedback-rating. To facilitate the differential local feedback-rating values to other interacted entities from the standpoint of a particular entity, and prevent a malicious entity from giving arbitrarily high feedback-ratings to other misbehaved entities, we need to normalize this local feedback-rating s_{ij}:

$$m_{ij} = \begin{cases} max(s_{ij},0)/\sum_j max(s_{ij},0) \; if \sum_j max(s_{ij},0) \neq 0 \\ p_j \quad otherwise \end{cases}, \tag{4}$$

where $P = \{p_j\}$ denotes the set of pre-trusted entities, $p_j = 1/|P|$ for j ∈ P and $p_j = 0$ otherwise.

2.2 Trust Propagation Kernel

The commonly used global trust aggregation fashion, such as EigenTrust [9], it is based on uniformly distributed trust propagation kernel, namely $\overrightarrow{T}^{(r+1)} = (1-u)M^T\overrightarrow{T}^{(r)} + u\overrightarrow{P}$, where $\overrightarrow{T}^{(r+1)}$ denotes all entities' global trust scores at the $(r+1)$th iteration round, $M = \{m_{ij}\}$ represents the matrix of normalized local feedback-rating and u is the probability that an entity knows none but relies on pre-trusted entities. Nevertheless, as analyzed in our previous work [4], this uniformly distributed trust propagation kernel suffers from several vulnerabilities to propagate trust along direct links from an entity to all its neighbors when the strategically misbehaved entities exist. Therefore, in this paper, on the basis of admitting the property of honesty of strategically misbehaved entities, we employ the affinity of two entities as a "bridge" to propagate belief/unbelief information instead of the direct local feedback-rating. Besides the consideration on connected neighbors of an entity, we also take those unconnected entities into account for the aggregation of marginal probability in our MapTrust.

3 MapTrust: Trustworthiness and Untrustworthiness Inference

3.1 Trustworthiness and Untrustworthiness Formulation

BP algorithm was initially proposed by Pearl [12] for solving the inference problems by passing local messages (belief) over diverse graphs, and successfully applied in error-correcting coding theory [14,15], social network [17] and biological networks [1], etc. Empirically, BP algorithm works surprisingly well even for graphical networks with loops [15], this property well fits the massive loops embedded interactional networks. In standard BP, a variable $h_{ij}(x_j)$ which can intuitively be understood as a "message" from a hidden node i to the hidden node j about what state node j should be in. The message $h_{ij}(x_j)$ will be a vector of the same dimensionality as x_j, with each component being proportional to know likely node i thinks it is that node j will be in the corresponding state. The belief at node i is proportional to the product of local evidence at this node $(\phi_i(x_i))$ and all the messages coming into i:

$$b_i(x_i) = z \cdot \phi_i(x_i) \prod_{j \in N(i)} h_{ji}(x_i), \tag{5}$$

where z is a normalization constant and $N(i)$ denotes the nodes neighboring i. It is easy to convince BP in fact gives the exact marginal probabilities for each nodes [14], and the messages can be formulated by the message update rule:

$$h_{ij}(x_j) = \sum_{x_i} \phi_i(x_i) \cdot \psi_{ij}(x_i, x_j) \cdot \prod_{k \in N(i) \setminus j} h_{ki}(x_i), \tag{6}$$

where $\psi_{ij}(x_i, x_j)$ denotes pairwise evidence between node i and node j. The right-hand side interprets the product over all messages going into node i except for the one coming from j.

Behind BP formula, two-layer connotations are included: (i) the belief going into node i from unconnected nodes (local evidence); (ii) the belief going into node i from directly connected nodes. To formulate this "message-passing", we denote several variables in the light of misbehavior-integrated interactional networks. Concretely, the multi-states are redefined as different groups to represent interacted behaviors' trustworthiness and untrustworthiness for different categories of entities. Assume the fraction of entities in each group a is n_a, the number of groups is q, correspondingly, a $q \times q$ affinity matrix can be defined to denote the probability P_{ab} over each edge between group a and group b. For a directed network, the adjacent matrix $A_{ij} = 1$ indicates there exists an edge (transactions) from entity i to entity j, otherwise no edge exists with $A_{ij} = 0$.

We first define the conditional marginal, denoted by $b_{g_i}^{i \to j}$, namely marginal/removal probability, implying entity i belongs to group g_i in the absence of entity j. Accordingly, we can compute the "message-passing" i sends j recursively in terms of the messages that i receives from other neighbors k [3,17]:

$$b_{g_i}^{i \to j} = \frac{1}{Z^{i \to j}} \cdot n_{g_i} \cdot \prod_{k \in N(i) \setminus j} \left[\sum_{g_k} P_{g_k g_i}^{A_{ki}} (1 - \frac{P_{g_k g_i}}{N})^{1 - A_{ki}} b_{g_k}^{k \to i} \right], \tag{7}$$

where $N(i)$ denotes the neighbors of entity i, $Z^{i \to j}$ is a normalization constant ensuring $\sum_{g_i} b_{g_i}^{i \to j} = 1$. Then, we can further define the marginal probability as:

$$b_{g_i}^{i} = \frac{1}{Z^{i}} \cdot n_{g_i} \cdot \prod_{k \in N(i)} \left[\sum_{g_k} P_{g_k g_i}^{A_{ki}} (1 - \frac{P_{g_k g_i}}{N})^{1 - A_{ki}} b_{g_k}^{k \to i} \right], \tag{8}$$

where Z^i is a normalization constant ensuring $\sum_{g_i} b_{g_i}^{i} = 1$. Formula (8) denotes the marginal probability entity i belongs to the group g_i.

For a pair of entities i and j, we can further discuss in two cases. If $A_{ij} = 0$, then we have (given $\sum_{g_k} b_{g_k}^{k \to i} = 1$):

$$b_{g_i}^{i \to j} = \frac{1}{Z^{i \to j}} \cdot n_{g_i} \cdot \prod_{A_{ki}=0}^{k \neq j} \left[1 - \sum_{g_k} \frac{P_{g_k g_i}}{N} \cdot b_{g_k}^{k \to i} \right] \cdot \prod_{A_{ki}=1} \sum_{g_k} P_{g_k g_i} \cdot b_{g_k}^{k \to i}. \tag{9}$$

If $A_{ij} = 1$, then we have:

$$b_{g_i}^{i \to j} = \frac{1}{Z^{i \to j}} \cdot n_{g_i} \cdot \prod_{A_{ki}=0} \left[1 - \sum_{g_k} \frac{P_{g_k g_i}}{N} \cdot b_{g_k}^{k \to i} \right] \cdot \prod_{A_{ki}=1}^{k \neq j} \sum_{g_k} P_{g_k g_i} \cdot b_{g_k}^{k \to i}. \tag{10}$$

Hence, we can rewrite the removal probability from entity i to entity j as (assume $b_{g_k}^{k \to i} = b_{g_k}^{k}$ if $A_{ki} = 0$):

$$b_{g_i}^{i \to j} = \frac{1}{Z^{i \to j}} \cdot n_{g_i} \cdot \prod_{A_{ki}=0} \left[1 - \sum_{g_k} \frac{P_{g_k g_i}}{N} \cdot b_{g_k}^{k} \right] \cdot \prod_{A_{ki}=1}^{k \neq j} \sum_{g_k} P_{g_k g_i} \cdot b_{g_k}^{k \to i}. \tag{11}$$

Correspondingly, the marginal probability for entity i can be redefined as:

$$b_{g_i}^{i} = \frac{1}{Z^i} \cdot n_{g_i} \cdot \prod_{A_{ki}=0} \left[1 - \sum_{g_k} \frac{P_{g_k g_i}}{N} \cdot b_{g_k}^{k} \right] \cdot \prod_{A_{ki}=1} \sum_{g_k} P_{g_k g_i} \cdot b_{g_k}^{k \to i}. \tag{12}$$

We define the marginal probabilities in different groups to mirror the fine-grained trustworthiness and untrustworthiness values with group assignment for each entity. Next, we interpret how to set transacted behavior-aware removal probability $b_{g_i}^{i \to j}$ for each pair of entities.

3.2 Removal Probability Setting

In misbehavior-integrated interactional networks, we can easily observe two scenarios: (i) good entities provide authentic services for other entities and give honest feedback-ratings to other entities; (ii) inversely, misbehaved entities provide inauthentic services and give dishonest feedback-ratings. Furthermore, for strategically misbehaved entities, they can provide authentic services occasionally in order to gain high positive feedback-ratings, but they always give dishonest feedback-ratings. Upon the above analysis, we utilize differential feedback behaviors to define pairwise removal probability based on the gravitation model:

$$b_{g_i}^{i \to j} = \begin{cases} \frac{m_{ij} \cdot m_{ji}}{d_{ij}^2} \; m_{ji} \neq 0 \\ \frac{m_{ij} \cdot \vartheta_{ji}}{d_{ij}^2} \; otherwise \end{cases}, \tag{13}$$

where $\vartheta_{ji} = 0$ if transactions happen between i and j and j as the service consumer, denoting j gives negative feedback-ratings to i; otherwise $\vartheta_{ji} = 0.5$ if no transaction takes place, implying they are strangers, the potential trustworthiness probability ought to be 0.5. Distance d_{ij} stands for the feedback-rating deviation to their common entities with which both i and j have had transactions. Generally speaking, the smaller the deviation is, the higher the affinity will be. Thus we define d_{ij} as:

$$d_{ij} = \left(\frac{\sum_{v \in cmn(i,j)} (m_{iv} - m_{jv})^2}{|cmn(i,j)|} \right)^{1/2}, \tag{14}$$

$comn(i,j)$ is the set of common entities transacted with both entities i and j.

The pairwise removal probability is the base for iteratively computing marginal probability, thus we need to initially ascertain which group the entity i belongs to by the affection of entity j. Concretely, we select some seminal trustworthy (pre-trusted) entities into a group (e.g. group a) as the initial members. Then, we define pairwise removal probability sequentially for each pair of connected entities in three cases as shown in Fig. 1, in which entities $e_1 - e_3$ are initially defined trustworthy members, and $e_4 - e_9$ are unknown entities: (i) two connected entities are initially trustworthy entities, e.g. edge (e_2, e_3), we define the removal probability as $b_a^{e_2 \to e_3} = m_{e_2 e_3} \cdot m_{e_3 e_2} / d_{e_2 e_3}^2$; (ii) one is

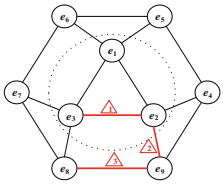


Fig. 1. Three cases of removal probability setting.

the initially trustworthy entity, the other is a unknown entity, e.g. edge (e_9, e_2). Given that e_2 had been dropped into group a, we define its removal probability $b_a^{e_9 \to e_2} = m_{e_9 e_2} \cdot m_{e_2 e_9} / d^2_{e_9 e_2}$; (iii) two connected entities are unknown entities, e.g. (e_8, e_9), we compute the maximum possibility which group e_9 is dropped into $\underset{q}{max}\{e_9\} = \underset{g_{e_9}}{max}\{b^{e_9 \to i}_{g_{e_9}}\}$. Upon this, we define its removal probability: $b^{e_8 \to e_9}_{\underset{q}{\max}\{e_9\}} = m_{e_8 e_9} \cdot m_{e_9 e_8} / d^2_{e_8 e_9}$. For the edges between initial seminal entities, it will be easy to define removal probabilities. For case (ii) it is also relatively easy to set removal probabilities referring to pre-selected group identity. Nevertheless, it would be hard to figure out removal probabilities in case (iii) since we need to ascertain an entity's maximum possibility dropped into a particular group. De facto, for a pair of unknown entities, we can archive removal probability setting via repeatedly traversal way.

We can assign many groups to represent different-level behavioral trustworthiness, such that literature [2] classified entities' behaviors into four continuums: Distrust, Undistrust, Untrust and Trust. Trust and distrust may not be derived from the same information but can coexist without being complementary [10,11]. Indeed, as our previously studied in [5], the strategically misbehaved entities can really gain high trust through uploading good services, and simultaneously get distrust occasionally through providing bad services. This is to say we cannot simply identify the strategic attackers as good or malicious entities, they hold both trustworthy and untrustworthy properties at the same time. Therefore, we focus on quantitatively inferring potential levels of trustworthiness and untrustworthiness simultaneously. To achieve this goal, we only need to set the number of groups as two in conformity with Formula (12).

Upon the defined marginal probability for each entity, we can alternatively compute the fraction of entities in each group:

$$n_a = \frac{1}{N} \sum_i b_a^i \tag{15}$$

Accordingly, the group affinity probability can be defined as:

$$\begin{aligned} P_{ab} &= \tfrac{1}{N} \cdot \tfrac{1}{n_a n_b} \cdot \sum_{A_{ij}=1} \tfrac{P_{ab}(b_a^{i\to j} \cdot b_b^{j\to i} + b_b^{i\to j} \cdot b_a^{j\to i})}{Z^{ij}} \\ Z^{ij} &= \sum_{a \neq b} P_{ab}(b_a^{i\to j} \cdot b_b^{j\to i} + b_b^{i\to j} \cdot b_a^{j\to i}) + \sum_{a=b} P_{aa} \cdot b_a^{i\to j} \cdot b_a^{j\to i}, \end{aligned} \tag{16}$$

where Z_{ij} is a normalization parameter. Obviously, we can work out the group affinity probability by power-law iteration. As aforementioned, the marginal probability computation also need employ power-iteration. Therefore, the entire trustworthiness and untrustworthiness inference is a cross-iteration process, and Algorithm 1 interprets how to accomplish this cross-iteration method.

Algorithm 1. Marginal Probability Computation.

Input: number of Groups q, maximum iteration round r_{max}, minimum iteration increment δ_{min}, current iteration round r, current increment δ
Output: marginal probability (trustworthiness/untrustworthiness) value b_q^i
{Group affinity probability initialization 3-5}
for each pair of groups a, b **do**
  Initiate $P_{ab}^{(0)}$ based on the rule: $P_{aa}^{(0)} > P_{ab}^{(0)}$ $(a \neq b)$
end for
{Trustworthiness initialization 6-8}
for each entity i $(i \in [1, n])$ **do**
  Initiate trustworthiness value for each entity $(b_{g_i}^i)^{(0)}$
end for
{Iteratively compute marginal probability 9-18}
while $r < r_{max}$ and $\delta > \delta_{min}$ **do**
  $r = r+1$
  for i=1 to n **do**
    Compute $(b_{g_i}^i)^{(r)}$ using Formula (12) via $b_{g_i}^{i \to j}$ and $P_{ab}^{(r-1)}$
  end for
  for each pair of groups a, b $(a, b \in [1, q])$ **do**
    Compute $P_{ab}^{(r)}$ according to Formula (16) using updated value $(b_{g_i}^i)^{(r)}$
  end for
  $\delta = |(b_{g_i}^i)^{(r)} - (b_{g_i}^i)^{(r-1)}| + |P_{ab}^{(r)} - P_{ab}^{(r-1)}|$
end while

4 Experimental Evaluation

4.1 Experiment Configuration

To evaluate the efficiency of our proposed MapTrust, we introduce four representative attack models commonly used in interactional networks: Independently Malicious (IM), Chain of Malicious Collectives (CMC), Malicious Collectives with Camouflage (MCC) and Malicious Spies (MS). For the detail definitions, please see references [5,9]. We compare MapTrust with randomly generated removal probability metric-Random and MCMC, a group assignment general algorithm. Table 1 depicts the configuration. A service requester first interacts with the candidates assigned in trustworthiness group. If more than one candidates exist, then utilize probabilistic selection fashion [5,9] to select transacted target. If no response entity, then give up the query and count a failure transaction.

Obviously, the higher the precision different categories of entities assigned appropriately, the more effective the algorithm. Thus, we define ***overlap*** between the original assignment $\{\mu_i\}$ and its grouped $\{q_i\}$ to evaluate the effectiveness:

$$A(\{\mu_i\}, \{q_i\}) = \frac{1}{N} \sum_i \delta_{\mu_i, \omega(q_i)}, \tag{17}$$

where ω ranges over the permutation on q groups with $\delta_{\mu_i,\omega(q_i)}$ being Kronecker delta, denoting if $\{\mu_i\}$ is equal to $\{q_i\}$, then $\delta_{\mu_i,\omega(q_i)} = 1$, otherwise $\delta_{\mu_i,\omega(q_i)} = 0$.

Table 1. Experimental parameters.

Experimental environment	Value
Number of entities in IM, CMC, MCC, MS	600, 600, 700, 1000
Number of pre-trusted entities	30
Initial neighbors of good, malicious and pre-trusted entities	2, 10, 10
Hops for query process	7
File distribution at good entities	Zipf distribution over 200 distinct files
Ratio of distinct files owned by good entities in IM, CMC, MCC	15%
Ratio of distinct files owned by good entities in MS	10%
Ratio of file types owned by malicious entities in IM, CMC, MS	100%
Ratio of file types owned by malicious entities in MCC	55%
Ratio of requests in which good entities give inauthentic file	5%
Probability that entities with global trust score 0 are selected	Interval [0%–10%]

4.2 Iteration Round Investigation

The convergence of BP algorithm is a complicated question [3,17], i.e. there are no firm results how many iteration rounds are exactly needed for the convergence. However, even the algorithm cannot converge, the messages can provide useful information after a certain number of iteration rounds [3]. With appropriate group number q, fraction of nodes in group n_a and affinity matrix P_{ab}, MapTrust will converge to a fixed point in a constant number of iteration rounds, otherwise it cannot converge. For MCMC algorithm, if the group number, group fraction and affinity matrix are not at the right values, its equilibration time

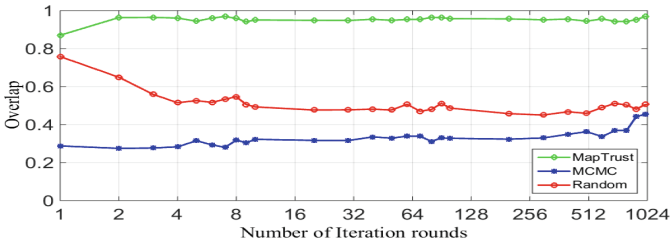


Fig. 2. Overlap with different iteration rounds.

diverges. Therefore, we study how iteration round affects the algorithm performance utilizing attack Model MCC with the probability f that attack entities provide good services is 0.0. The experimental results are shown in Fig. 2.

We see MapTrust maintains a high overlap value with a subtle vibration as iteration round enlarges, which indicates it can approach an ideal performance and become stable within a small number of iteration rounds, e.g. 10. Nevertheless, MCMC keeps a climbing status as iteration round increases, and approaches 70% approximately at a large number of iteration rounds, e.g. 1024. For Random, it makes a little variation and meets the practical case after a few iterations, e.g. 20. Therefore, this group of experiments verifies that MapTrust produces an appropriate performance within a small number of iteration rounds, while MCMC needs a large number to make the performance towards a better tendency. In the following experiments, we run MapTrust, MCMC and Random algorithms using 10, 1024 and 20 iteration rounds respectively.

4.3 Group Affinity Probability

Group affinity plays an important role for the group assignment and calculation of trustworthiness and untrustworthiness values. We here study how this group affinity probability affects the assignment of different types of entities. We utilize MCC to perform experiments wherein the probability f is 40%. One big goal is to identify authentic entities from collusively misbehaved entities, assign them into different groups. Accordingly, we set the total group number as two and define inter-group affinity probability from the viewpoints of two primary intervals as (0.0, 0.5] and (0.5, 1.0], in each range, we again set small intervals. The experimental results are depicted in Fig. 3.

We can observe the intra-group affinity does not affect the overlap on the whole. This indicates our proposed cross-iteration algorithm does not need particular intervals for intra/inter-group affinity to achieve an adequate performance. Furthermore, this also verifies the rationality and correctness of our cross-iteration fashion. In following sections, we do not restrain intra/inter-group affinity probability and arbitrarily generate a decimal.

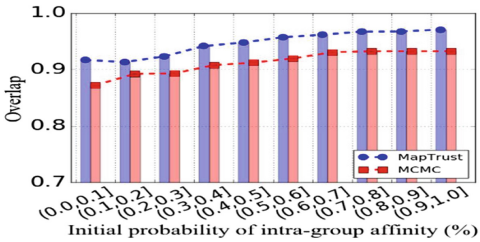


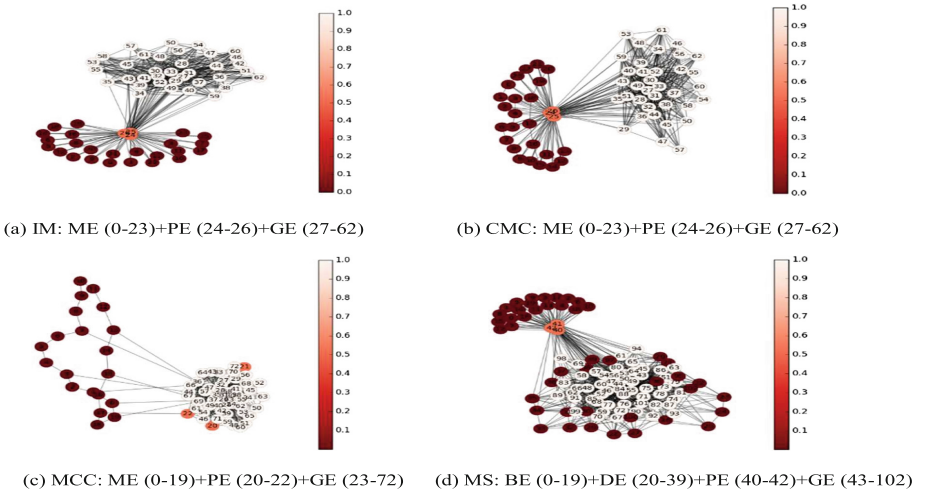
Fig. 3. Overlap with initial intra-group affinity probability.

4.4 Performance with Synthetic Datasets

We launch a set of small-scale experiments, i.e. 63 entities included in IM and CMC with percentage of malicious entities 40%, 73 entities contained in MCC with 20 camouflage entities and $f = 0.4$, 103 entities exist in MS with 20 type D and 20 type B. The numbers of transactions are 630 in IM and CMC, 730 in MCC and 1030 in MS. The results are depicted in Fig. 4, in which the fine-grained trustworthiness and un-trustworthiness values are exhibited with gradient color for each entity. "ME", "PE", "GE", "BE" and "DE" denote misbehaved, pre-trusted, good, type-B and type-D entities respectively. Obviously, good and malicious entities are clustered into two groups with quantitative trustworthiness and untrustworthiness values. In addition, we increases percentage of misbehaved entities from 0% to 70% under IM and CMC, vary f from 0% to 80% under MCC, and change combinations of type D and type B entities under MS. The results are depicted in Fig. 5.

From Fig. 4, we can observe MapTrust clearly clusters the good and misbehaved entities into two groups with fine-grained trustworthiness and untrustworthiness values for each entity. From Fig. 5, we can observe MapTrust significantly outperforms MCMC and Random. The overlap of MapTrust is almost optimal in IM and CMC, i.e., the overlap values are all 1.0 from 40% and 30% in IM and CMC, which demonstrate MapTrust is good at handling the worse scenarios. Nevertheless, the overlap in MCMC declines from 0.81/0.84 to 0.32/0.32 in IM/CMC as misbehaved entities increase. The overlap in Random goes up from 0.50/0.55 to 0.75/0.77 accordingly. The behind reason lies in that MCMC only utilizes the energy increase to adjust when to move one entity into another group ignoring the pairwise removal probability setting, thus it performs worse while confronting isolated and collusive misbehaviors, the performance becomes worse as the ratio of malicious entities enlarges. For Random, since the pairwise removal probability over each edge is set randomly without considering the feedback-rating feature, thus it must be worse than MapTrust.

In MCC and MS, MapTrust also yields significant results, e.g. the overlaps are between [0.87, 0.93] and [0.85, 0.91]. In MCC, the misbehaved entities act as good ones with probability f, thus some malicious entities are misidentified as good entities. The overlap produced by MCMC is of a little variation as the f increases, this is because the group affinity will not change dramatically in



(a) IM: ME (0-23)+PE (24-26)+GE (27-62)

(b) CMC: ME (0-23)+PE (24-26)+GE (27-62)

(c) MCC: ME (0-19)+PE (20-22)+GE (23-72)

(d) MS: BE (0-19)+DE (20-39)+PE (40-42)+GE (43-102)

Fig. 4. Trustworthiness and untrustworthiness values with group assignment. (Color figure online)

the case of only varying f. The overlap produced by Random also has a subtle change, this is also because the subtle variation of group affinity probability. In MS, type D entities behave alike good entities to promote all type B entities, thus it is a little hard to completely identify them. The overlap yielded by MCMC is almost changeless while confronting various combinations of type B and type D entities, this is because the group affinity probability will not change obviously in the case of only varying the combination manners, but not the ratio of malicious entities. Random also has a subtle variation owing to the same reason. Although facing the collective camouflage and spy entities, our MapTrust still can achieve much better than MCMC and Random.

4.5 Performance with Real-World Datasets

We also utilize real-world dataset Epinions [13] to evaluate the effectiveness of our MapTrust in terms of attack resilience and computational complexity. MCC and MS are adopted to evaluate attack resilience with an alike configuration, i.e., 10, 30 and 50 strategically misbehaved entities are added into 100-entity organized Epinions network. These misbehaved entities are connected to most of entities with high degrees to receive as many feedback-ratings as possible. For MCC, all the added misbehaved entities which compose a chain, own a certain probability f to response good services to gain positive feedback-ratings, then in return to give exaggerated feedback-ratings (1.0) to their partners. For MS, the misbehaved entities play two roles: type D and type B. Type D entities acts as good ones to provide good services to gain high feedback-ratings, then in return give high feedback-ratings (1.0) to all type B entities. Therefore, an interval [0.85, 1.0] is used for the setting of feedback-ratings from good entities

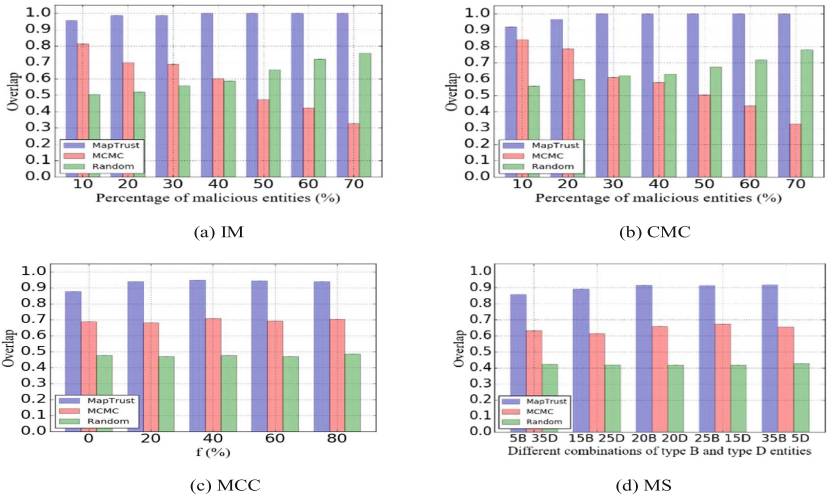


Fig. 5. The performance under four attacks with synthetic datasets.

to all the camouflage and spy entities, another interval [0, 0.05] for the setting of feedback-ratings from the strategically malicious entities to good ones.

Figure 6 shows the experimental results, where "100GE+10ME" represents the network contains 100 regular (good) and 10 added (misbehaved) entities, "100GE+5BE+5DE" denotes 100 good, 5 type B and 5 type D entities. We can see MapTrust performs much better than MCMC and Random. For MCC with variable f, the overlap in MapTrust can keep a high level, but MCMC makes a subtle variation with poor performance. Random performs worst and declines the overlap a little bit as the f increases. In the case of varying the amount of misbehaved entities, MapTrust and MCMC can maintain the results changeless, but Random groups good and malicious entities more correctly. For MS with different combinations of type B and type D entities, MapTrust, MCMC and Random keep a relatively steady performance. At the case of varying the number of malicious entities, MapTrust declines as the amount of misbehaved entities increases, but Random inversely can improve the overlap gradually. Even so, our proposed MapTrust still dramatically outperforms MCMC and Random.

From Formula (12), we know the marginal probability for each entity is computed by aggregating all removal probabilities between this entity and connected neighbors, in addition to the consideration on other unconnected entities, the update computation needs to ask the other $(N-1)$ entities' information. Therefore, it takes $O(cN)$ time overhead totally, c is the average degree. This indicates our MapTrust takes linear time in the common case of a sparse large-scale collaborative networks. For MCMC, it always needs to compute the energy difference by adopting group affinity of two entities over all the edges cN, thus it takes $O(cN^2)$, however, in practice we only choose those edges that are connected to the candidate entity i, which might be removed into a new group or

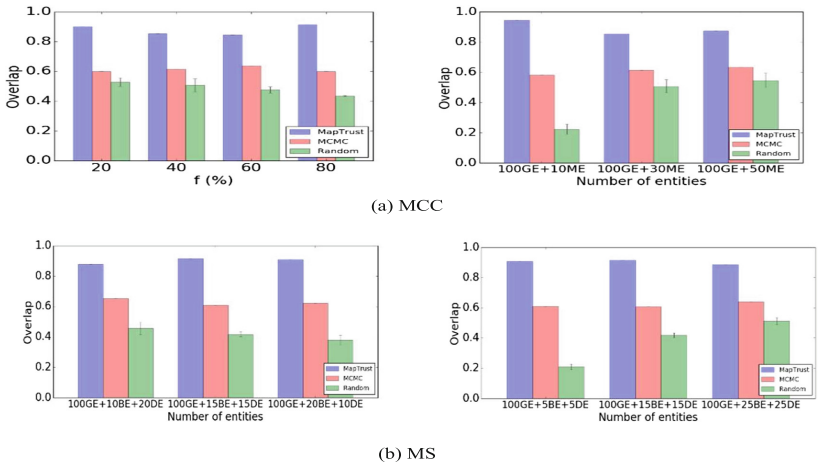


Fig. 6. The performance under MCC and MS with Epinions datasets.

not, to compute the energy difference. Thus, the energy difference computation just needs to choose the edge(s) that connected to i, thus it needs to ask all the neighbors c_i. Therefore, the time overhead for all the entities takes $O(cN)$ totally.

5 Related Work

Gaeta and Grangetto [6] proposed to use BP to assess the probability of an entity being malicious in P2P streaming, in which a set of pre-trusted monitors are utilized to check the chunk uploaders and mark the chunk as polluted or clean. This work focuses on the prediction whether an entity is malicious. However, our work aims at employing marginal probabilities to estimate the multi-probabilities representing the possibilities one entity belongs to different groups, simultaneously infer its fine-grained trustworthiness and untrustworthiness values.

Zhang et al. [17] proposed a core-periphery structure identification algorithm on empirical network through the application of an expectation-maximization (EM) algorithm for the parameter computation and BP algorithm. Decelle et al. [3] also utilized BP algorithm to infer functional groups by maximizing the overlap with the potential group members, and learn the unknown parameters of the block model, in which diverse entities are clustered into different groups based on topology structure. However, our MapTrust is inspired partially by the above work [3,17], but differently we mainly focus on the problem of resisting various isolated/collusive attacks by grouping them into adequate clusters and calculate their different-level trustworthiness and untrustworthiness values, but not for the exploration on network structure/topology merely. Moreover, we redefine pairwise removal probability for each pair of interacted entities and endow new content for trust-enabled interactional networks.

6 Conclusion and Future Work

We have stated our trustworthiness and untrustworthiness inference algorithm with group assignment through studying pairwise removal probability and marginal probability. We also verify the efficiency of our MapTrust through extensive experiments using synthetic and real-world datasets. The experimental results show our proposed MapTrust not only appropriately groups good and misbehaved entities under the four representative attack models, but it can also rationally calculate the trustworthiness and untrustworthiness values in a fine-grained way for diverse entities. At present, our work mainly takes into account trustworthiness and untrustworthiness inference through clustering all participants into two groups, in the future we can explore a fine-grained solution to divide diverse categories of participants into more than two groups appropriately with respect to differentially malicious behaviors, such as independently/collectively malicious behaviors, camouflage and spy behaviors, etc.

References

1. Bailly-Bechet, M., Borgs, C., Braunstein, A., Chayes, J., Dagkessamanskaia, A., Francois, J.M., Zecchina, R.: Finding undetected protein associations in cell signaling by belief propagation. PNAS **108**(2), 882–7 (2011)
2. Cho, J.H., Chan, K., Adali, S.: A survey on trust modeling. ACM Comput. Surv. **48**(2), Article 28, 40 p. (2015)
3. Decelle, A., Krzakala, F., Moore, C., Zdeborová, L.: Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. Phys. Rev. **E84**, 066106 (2011)
4. Fan, X., Liu, L., Li, M., Su, Z.: EigenTrust^{++}: attack resilient trust management. In: Proceedings of 8th IEEE International Conference on Collaborative Computing, pp. 416–425. IEEE (2012)
5. Fan, X., Liu, L., Li, M., Su, Z.: Grouptrust: dependable trust management. IEEE Trans. Parallel Distrib. Syst. **28**(4), 1076–1090 (2017)
6. Gaeta, R., Grangetto, M.: Identification of malicious nodes in peer-to-peer streaming: a belief propagation-based technique. IEEE Trans. Parallel Distrib. Syst. **24**(10), 1994–2003 (2013)
7. Hu, H., Lu, R., Zhang, Z., Shao, J.: REPLACE: a reliable trust-based platoon service recommendation scheme in VANET. IEEE Trans. Veh. Technol. **66**(2), 1786–1797 (2017)
8. Jøsang, A., Ismail, R.: The beta reputation system. In: Proceedings of the 15th Bled Electronic Commerce Conference, vol. 160, pp. 41–55 (2002)
9. Kamvar, S., Schlosser, M., Garcia-Molina, H.: The EigenTrust algorithm for reputation management in P2P networks. In: Proceedings of the 12th International Conference on World Wide Web, pp. 640–651. ACM (2003)
10. Luhmann, N.: Familiarity, confidence, trust: problems and alternatives. Trust Mak. Peaking Coop. Relat. **6**, 94–107 (2000)
11. Marsh, S., Dibben, M.R.: Trust, untrust, distrust and mistrust – an exploration of the Dark(er) side. In: Herrmann, P., Issarny, V., Shiu, S. (eds.) iTrust 2005. LNCS, vol. 3477, pp. 17–33. Springer, Heidelberg (2005). https://doi.org/10.1007/11429760_2

12. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, Burlington (1988)
13. Richardson, M., Agrawal, R., Domingos, P.: Trust management for the semantic web. In: Fensel, D., Sycara, K., Mylopoulos, J. (eds.) ISWC 2003. LNCS, vol. 2870, pp. 351–368. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39718-2_23
14. Yedidia, J.S., Freeman, W.T., Weiss, Y.: Constructing free-energy approximations and generalized belief propagation algorithms. IEEE Trans. Inf. Theory **51**(7), 2282–2312 (2005)
15. Yoon, S., Goltsev, A.V., Dorogovtsev, S.N., Mendes, J.F.: Belief-propagation algorithm and the ising model on networks with arbitrary distributions of motifs. Phys. Rev. **E84**, 041144 (2011)
16. Su, Z., Liu, L., Li, M., Fan, X., Zhou, Y.: Reliable and resilient trust management in distributed service provision 5networks. ACM Trans. Web **9**(3), Article 14, 37 p. (2015)
17. Zhang, X., Martin, T., Newman, M.E.J.: Identification of core-periphery structure in networks. Phys. Rev. **E91**, 032803 (2015)