Capability-aware Trust Evaluation Model in Multi-agent Systems

Tung Doan Nguyen, Quan Bai, and Weihua Li

School of Engineering, Computer and Mathematical Sciences,
Auckland University of Technology, New Zealand,
{tung.nguyen, quan.bai, weihua.li}@aut.ac.nz

Abstract. Modeling trust in a real time of dynamic multi-agent systems is important but challenging, particularly when agents frequently join and leave, and the structure of the society may often change. With the increasing complexity of services, some simplified assumptions, e.g., unlimited processing capability, adopted by several trust models have shown their limitations which restrict the application of trust model in real-world situations. This paper attempts to relax the unlimited processing capability assumption of agents by introducing a capability-aware trust evaluation with temporal factor using hidden Markov model. The experimental results show that the approach not only can improve the accuracy of trust computation but also benefit the trust-aware decision making for both individual and agent group context.

Keywords: multi-agent system, trust, composite services, capability-aware

1 Introduction

In open, dynamic environments of Multi-Agent Systems, agents have high possibility to be exposed to risk when interacting with strangers. Trust has become an essential tool for selecting interaction partners by reducing the uncertainty of interactions, promoting robustness and vitality of diverse social interactions [8]. Several computational trust models have been proposed to address different situations. Many of them assume agent to be rational with unlimited processing capability (UPC), i.e., the performance of agents is not affected by the number of requests [11]. However, the survey [3] also points out that the trustee capability is closely related to timeliness of task completion. It is part of the quality metric for truster agents. The practice has shown the defect of not including agents' capability into trust model [9, 11]. The issue exists in many popular trust models, for example, [10, 4, 2]. It can also lead to reputation damage problem [13].

Recently, the context-aware trust evaluation has been receiving a great attention [5, 6]. In [5], the authors tried to connect the feature set of a trustee agent to its performance for computing the trust value. The accuracy of the approach depends on training data and it is not suitable for distributed environment. Taking limited capability into account, the work of Yu [11, 12] benefits the trustee agents by improving the request management. However, the approach does not address the quality of trust evaluation for truster agents.

In many practical applications, e.g. composite services, trustee agents can group up to deliver more sophisticated services to truster agents. Single-tasking is not preferable in a group context since it significantly reduces the productivity the group. For example, when one agent is processing requests, another agent can process other tasks rather than waiting. Since the quality of service of a group is closely related to the agent coordination, it is better to consider the agents' capability in trust model. From the perspective of truster agents, single or multi-tasking model of a service is uncertain to most of them. Truster agents do not have to know internal activities within a group to evaluate its trustworthiness. Nevertheless, considering agent or group of agents as an entity with limited processing capability can have a significant impact on the evaluation process. Because the current state (resource states) of a service provider can be used to improve the accuracy of the trust evaluation.

In this paper, we try to address the relation between the number observable requests and the outputs with trustee's capability. So how can we take the agents capability into the trust evaluation model? We find that the hidden Markov model (HMM) can be used to reveal the relation between the temporal states and the capabilities of agents, therefore, can be used for computing trust values and relax the UPC assumption. This paper firstly tries to determine capabilities of agents and then evaluates their trustworthiness based on HMM.

2 Definitions

This work assumes that trustee agents or groups can accept multiple requests while processing other tasks (see Figure 1). The number of requests can impact the output, i.e., Quality of Service (QoS). A trustee agent delegates a task not only based on reputation value, but also current situation of each trustee candidate based on our proposed trust evaluation method.

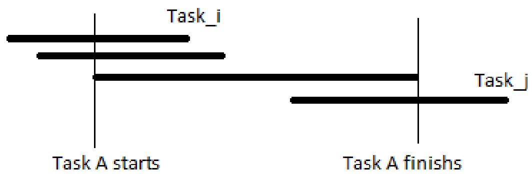

Fig. 1. Tasks handling of a trustee agent

Definition 1. *The* ***capability-aware trust*** $T_{(c_i,p_j)}$ *represents the evaluation of truster* c_i *over the trustworthiness of trustee* p_j*. It is a real value indicating the probability that trustee* p_i *will produce an expected output when* p_j *is processing* k *concurrent tasks, i.e.,* $T_{(c_i,p_j)} = P(QoS_{d_{p_j}} \geq QoS_{r_{c_i}} | CT_{p_j} = k)$.

$QoS_{r_{c_i}}$ and $QoS_{d_{p_j}}$ are the expected QoS of c_i and the QoS of delivered service of p_j, respectively. Trust is considered to be subjective; different truster agents can have different trust values over the same trustee agent. Thus, this paper models trust as a conditional probability of producing the expected output when given the certain number of requests. The trust value is in 0 and 1, which stands for the most untrustworthy and the most trustworthy values, respectively.

Definition 2. *The **historical record** H_{p_i} of agent p_i is a set of transaction (see Definition 4) records, i.e., $H_i = \{tran_{i_1}, tran_{i_2}, tran_{i_3}, \dots\}$, which is kept at the local database of agent.*

Each transaction contains public verifiable information from starting request to rating stage. This assumption can boost the confidence of evidence collecting, especially in distributed environments.

Definition 3. *A **request** or a task is a contract between a truster c_i and a trustee p_j respecting a service. A task can be represented by a double $req_{c_i,p_j} = \{c_i, p_j, st, t_{dl}\}$, where st is service type and t_{dl} is the deadline of the task.*

Definition 4. *The **transaction record** is data of one transaction session, represented by a 3-tuple $tran_{(c_i,p_j)} = \{t_r, t_d, r_{c_i,p_i}\}$, where t_r, t_d , r_{c_i,p_i} represent the timestamp of the request, timestamp of the delivered service, and the rating of c_i (see Definition 5), respectively.*

Definition 5. *The **rating** r_{c_i,p_i} is a binary value of either 0 or 1, indicating the unsatisfied and satisfied evaluation of the output of p_i, respectively.*

Since the task includes both the QoS and deadline, truster agents satisfy when both conditions are met:

$$r_{c_i,p_j} = \begin{cases} 0 & \text{if } QoS_{r_{c_i}} > QoS_{d_{p_j}} \text{ or } t_d > t_{dl} \\ 1 & \text{if } QoS_{r_{c_i}} \leq QoS_{d_{p_j}} \text{ and } t_d \leq t_{dl} \end{cases} \quad (1)$$

3 The Capability-aware Trust Evaluation

To take the capability into account, this paper adopts the following assumptions: (1) The requests for services from different trustee agents are observable by other trustee agents; (2) The historical records of agents are visible and verifiable; (3) The internal activities of an agent (or agent group) are unknown to truster agents, all coordination are handled privately by service providers.

3.1 Capability measure of trustee agents

To find the capability of a trustee agents, this paper utilizes the available information obtained from historical transactions including ratings and timestamps. A truster agent can observe the state of a trustee agent at the time of making request and the time of delivering service. A truster c_i can calculate the number of concurrent tasks (CT) of trustee p_i when accepting a request at the time tr_k:

$$CT_{tr_k} = 1 + \left|\{req_m : tr_k \in [tr_m, td_m]\}\right| \quad (2)$$

, where req_m is a request of the m^{th} transaction of H_{p_i}; td_m is the delivery timestamp of the m^{th} transaction. Because the CT is bounded, i.e., $0 < CT_{tr_n} \leq Max\{CT_{tr_n}\}$, we can use the Bayesian discrete probability [1] to estimate the probability of getting good output given the number of concurrent tasks. We compute the posterior probability according to Bayes' theorem:

$$P(r = 1 \mid CT = k) = \frac{P(CT = k \mid r = 1) \cdot P(r = 1)}{P(CT = k)} \quad (3)$$

, where prior probability $P(r = 1)$ is the probability of $r = 1$ before $CT = k$ is observed. $P(r = 1 \mid CT = k)$ is the posterior probability, is the probability of $r = 1$ given $CT = k$, i.e., after $CT = k$ is observed. The prior probability $P(CT)$ can be calculated by using Equation 4:

$$P(CT = k) = \frac{\left|\{tran_j \in p_i : CT = k\}\right|}{N} \quad (4)$$

, where N is total number of transactions in historical records. From the calculated CT_k with the associated outcome of the k^{th} transaction, we have a set of prior probability. $P(r \mid CT)$ can be interpreted as for a given number of concurrent tasks, the probability of the outcome r is $P(r \mid CT)$.

3.2 Capability-aware trust value calculation

To calculate the capability-aware trust value, firstly, the paper estimates the capability of a trustee from collective of evidence. We also use the Δ_t denotes the forgetting factor for outdated evidence, $\delta = \lambda^{\Delta_t}$. Having $\lambda = 1$ is equivalent to not having Δ_t, i.e., nothing is forgotten. Whereas, having $\lambda = 0$ results in only the last feedback value to be counted and all others to be completely forgotten. This time discount factor can assure that the outdated will be omitted from evaluation. A valid transaction evidence satisfies the following:

$$tran_i = \begin{cases} 1 & \text{if } \delta \geq 0.5 \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

, where 0 and 1 represent invalid and valid transaction record to use. As truster agents can observe the state of trustee agent the time of making the request but can not sure about the result at the time of delivering, we can model the trust evaluation using HMM with the memory of size 1 (first-order Markov model)[7].

Figure 2 shows the model of HMM of current problem using Trellis diagram. To a truster agent c_i, state S_k of trustee p_j is the number of $(k-1)$ concurrent tasks when a new request is made. $x = \{x_1, x_2, ..., x_n\}$ is a sequence of

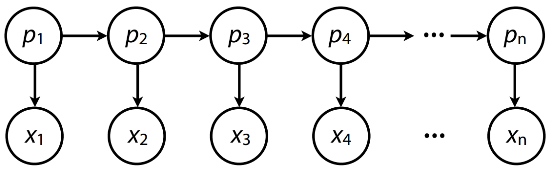

Fig. 2. HMM for capability-aware trust evaluation

observed output. Like Markov chain, edges capture conditional independence. For example, x_2 is conditionally independent of everything else given s_2.

As mentioned in Subsection 3.1, the probability of the output is based on the processing requests of trustee agent p_i. We can incorporate the probability with a two-step trust evaluation. Firstly, it uses a traditional approach to evaluating the trustworthiness of an agent based on the rating evidence. Secondly, the value will be adjusted based on the situational evaluations by considering the capability and current processing tasks.

Suppose that at the time of evaluation, trustee agent p_j has k ongoing requests. The number of concurrent tasks will increase by one if p_j accepts the request from c_i. In Subsection 3.1, trustee agent c_i has the table of transition state of p_j at time "$t+1$" (see Table 1).

Table 1. Transition table for states of trustee agent

	s_i	s_j	s_k
s_i	0.2	0.6	0.2
s_j	0.3	0.1	0.6
s_k	0.5	0.3	0.2

	$r=0$	$r=1$
s_i	0.3	0.7
s_j	0.6	0.4
s_k	0.2	0.8

The problem can be described as evaluation problem of HMM, given the observation sequence x and a formal HMM

$$\lambda = (A, B, \pi) \tag{6}$$

, where A is a transition array, storing the probability of state j following state i.

$$A = \{p_{ij} | p_{ij} = P(CT_{t+1} = j | CT_t = i)\} \tag{7}$$

and B denotes the observation array, storing the probability of observation k being produced from the state j, independent of t:

$$B = \{bi(k)|, bi(k) = P(xt = vk | qt = si)\} \tag{8}$$

How do we compute $P(x|\lambda)$, i.e., the probability of the observation sequence given the model. We can use the equation below to calculate the probability of the observations x for a specific state sequence state S:

$$P(x|Q, \lambda) = \prod_{t=1}^{P} (x_t|q_t, \lambda) \tag{9}$$

Finally, we can calculate the probability of the observation given the model as:

$$P(O|\lambda) = \sum_{Q} P(O|Q, \lambda) P(Q|\lambda) \tag{10}$$

However, due to the fact that our model is first-order HMM. So we can obtain the capability-trust evaluation with:

$$T(c_i|S_{p_j} = k) = \sum p(S_i|S_k) p(r = 1|S_i) \tag{11}$$

4 Experimental Results

The experiment evaluates the trust values of providers with different timestamps and states. However, we use agent groups in the experiments for better simulating the multi-tasking activities. Ten composite services with different profiles were created. All groups provide the same service type to simplify the experimental settings. We compared the approach with a traditional probability-based trust evaluation in two different scenarios: (1) trustee groups with a handleable number of requests, and (2) trustee groups with an excessive number of requests.

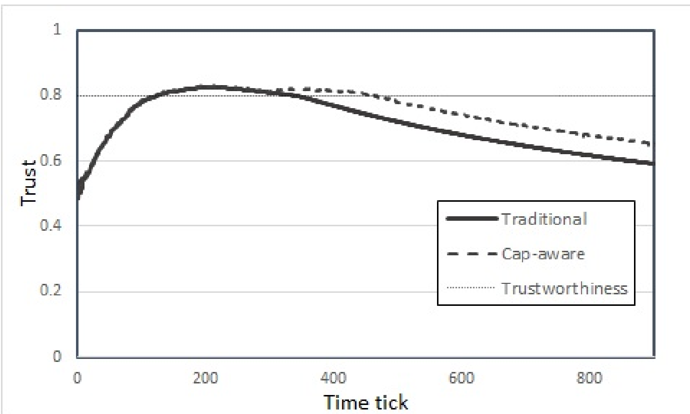

Fig. 3. Capability-aware VS. a traditional trust values

In the first experiment, the initial phase set agents with some transactions; the number of requests at a time was set to be handleable. We then monitor the trust value of each entity in case of no incoming request to see the effect of the temporal factor and trust. Figure 3 shows the result of two approaches in two phases. At first, under the handleable number of requests, both trustee groups produce the expected outputs. Thus, there is no significant difference between ours and the traditional approach. However, when there is no incoming request, the HMM can keep the trust value relatively close to the real trustworthiness.

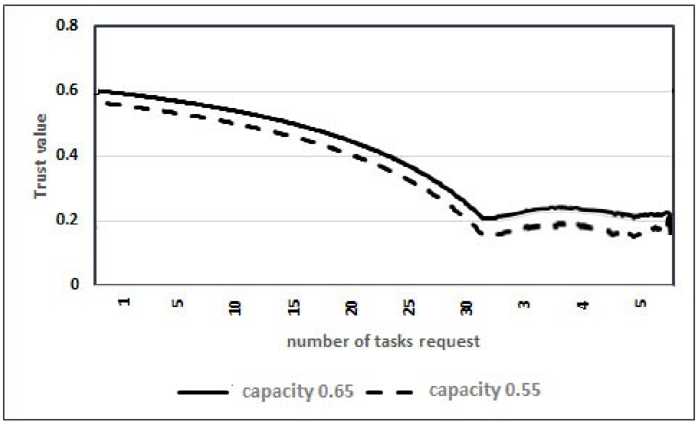

Fig. 4. Capability-aware trust value with varied number of requests

In the second experiment, we varied the number of requests each time step by increasing and then decreasing the number to see the effect of the model. Figure 4 illustrates the trust values of a truster agent observes two different trustee agents with different capability profiles. The results show that when the number of concurrent tasks exceeded the handleable capability, the trust values to both agents decreased, and the one with higher capability are perceived as more trustworthy. Interestingly, when the number of requests decreased to a handleable value, the trust value started to increase. This perfectly demonstrates the effect of the visible requests on capability-aware trust.

The above results also show the intuition that in practical situations, the reputation may be decreased because of performance affected by the excessive requests. However, the trust value for the trustee starts to increase when the completed tasks are released. Namely, lower reputation agents do not always mean lower expectation of the output.

5 Conclusions

The paper proposes a trust model considering agents' limited capability to relax the UPC assumption by using HMM approach. The model can produce promising results of trust evaluation for truster agents at the time of making requests. Our future work is set to improve the experiments and discover different QoS influencing factors for the model rather than using only the number of requests.

References

1. A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian data analysis*, volume 2. Taylor & Francis, 2014.
2. T. D. Huynh, N. R. Jennings, and N. R. Shadbolt. An integrated trust and reputation model for open multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 13(2):119–154, 2006.
3. A. Josang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618 – 644, 2007. Emerging Issues in Collaborative Commerce.
4. A. Jsang and J. Haller. Dirichlet reputation systems. In *Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference on*, pages 112–119, April 2007.
5. X. Liu and A. Datta. Modeling context aware dynamic trust using hidden markov model. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.
6. M. Naseri and S. A. Ludwig. *Data Provenance and Data Management in eScience*, chapter Evaluating Workflow Trust Using Hidden Markov Modeling and Provenance Data, pages 35–58. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
7. L. R. Rabiner and B.-H. Juang. An introduction to hidden markov models. *ASSP Magazine, IEEE*, 3(1):4–16, 1986.
8. J. Sabater and C. Sierra. Social regret, a reputation model based on social relations. *SIGecom Exch.*, 3(1):44–56, dec 2001.
9. S. Sen. A comprehensive approach to trust management. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems*, AAMAS '13, pages 797–800, Richland, SC, 2013. International Foundation for Autonomous Agents and Multiagent Systems.
10. Y. Wang and M. P. Singh. Evidence-based trust: A mathematical model geared for multiagent systems. *ACM Trans. Auton. Adapt. Syst.*, 5(4):14:1–14:28, Nov. 2010.
11. H. Yu, C. Miao, B. An, C. Leung, and V. R. Lesser. A reputation management approach for resource constrained trustee agents. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pages 418–424. AAAI Press, 2013.
12. H. Yu, C. Miao, B. An, Z. Shen, and C. Leung. Reputation-aware task allocation for human trustees. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*, AAMAS '14, pages 357–364, Richland, SC, 2014. International Foundation for Autonomous Agents and Multiagent Systems.
13. H. Yu, Z. Shen, C. Leung, C. Miao, and V. Lesser. A survey of multi-agent trust management systems. *Access, IEEE*, 1:35–50, 2013.