Southeast Asian Journal of Sciences: Vol 1, No. 2 (2012) pp. 148-154

CLASSES OF TRUST FUNCTIONS FOR DISTRIBUTED INTELLIGENT COMPUTING

Dinh Que Tran, Manh Hung Nguyen
and
Viet Phuong Ngo

[*] *Department of Information Technology*
Posts and Telecommunications Institute of Technology (PTIT)
Km10, NguyenTrai, Hadong, Hanoi, Vietnam
e-mail: tdque@yahoo.com, quetd@ptit.edu.vn , nmhufng@yahoo.com

[†] *Vietnam Korea Friendship Information Technology College*
Danang, Vietnam
vietphuong@mic.gov.vn

Abstract

Trust plays a crucial role for interaction among autonomous entities in distributed intelligent computing. In this paper, we first present various models of computational trust and describe the formal definitions of the concept. Then we investigate classes of computational trust functions which has been constructed from various trust types. In addition, we consider a class of special functions that represent the reference trust from lying entities.

1 Introduction

Trust has been widely investigated in various view points from sociologists, psychologists, economists and computer scientists. It has been computational models for recommender systems in various applications such as E-Business, Multi-Agent Systems, Web Services, Grid Computing, P2P networks, Adhoc

Key words: trust, reputation, inference, distributed intelligent computing.

networks etc. Most of the computational models ([1]–[11]) have accepted the definition of trust as a quantified belief by a *truster* on a *trustee* with respect to the competence, honesty, security and dependability within a specified context. However, rather than the formal investigation, the current researches in computational trust focus on construction of algorithms with various types of trust: (i) *Experience trust:* computing trust based on truster personal experience; (ii) *Reference trust:* estimating trust value by using evaluation feedback of partners in the community. This type of trust is also named reputation; (iii)*Similarity trust:* estimating trust value based on similarity of partners on profile or evaluation in the community; (iv) *Unreliable trust:* measuring trust value from partners who may provide unreliable information; (v) *Hybrid trust:* using hybrid combination of some trust types. To our knowledge, most of hybrid models concentrate on only considering the integration of experience and reference trusts. Combining more trust resources for computation needs to be considered furthermore.

In this paper, rather than focusing on computational aspect, we describe a mathematical approach to formalize the construction of trust concepts from various viewpoints and then investigate classes of functions contributing to the construction of computational algorithms. The remainder of this paper is structured as follows. Section 2 presents preliminaries for developing the mathematical models. Section 3 describes definitions for trust types and functional classes for contributing to construction of computational models of trust. Conclusion is presented in Section 4.

2 Preliminaries

We call a software agent or briefly agent to be an autonomous entity who takes part in computational cooperation of a distributed intelligent system. This section presents the definitions and notations which are basis for the presentation in the next section.

- $\mathcal{A} = \{a_1, \cdots, a_n\}$ be a set of agents, representing providers or consumers of items or services;
- U_{ij} is a set of interaction among agents i and j according to the time and then $\mid U_{ij} \mid$ is a number of interaction;
- $\mathcal{W}_i \subset \mathcal{A}$ is a world of agents, with whom agent i has interacted in the past;

Definition 1. *A trust is a function $t : \mathcal{A} \times \mathcal{A} \rightarrow [0, 1]$, in which $[0, 1]$ is an unit interval of the real numbers. The value $t(a, b) = \alpha$ means that a (truster) trusts b (trustee) with respect to the degree α.*

3 Trust Function Classes

3.1 Class of Experience Trust Functions

Intuitively, the evaluation in trust of an agent a on an agent b depends on temporal interaction among them. It means that the dynamics of trust are represented by means of the change of trust values in time. For simplicity, we denote $\mathcal{T}$ to be the set of temporal points and $w = (w_1, w_2, ...w_{|U_{ij}|})$ to be the weight vector, in which $| U_{ij} |$ is a number of interaction among agents i and j, such that $\sum_{k=1}^{|U_{ij}|} w_k = 1$ and $w_{k_1} \geqslant w_{k_2}$ if $k_1 < k_2$. The vector is used to describe the weighted evaluation of temporal trust. The constraints $w_{k_1} \geqslant w_{k_2}$ if $k_1 < k_2$ illustrates a fact that the more recent evaluation of an agent i on an agent j is more important than the previous ones.

Definition 2. *([9]) Experience trust of agent i in agent j is a mapping e_{ij} : $[0,1]^{|U_{ij}|} \times [0,1]^{|U_{ij}|} \to [0,1]$, which is defined by the formula*

$$e_{ij}(t_{ij}, w) = t_{ij} * w = \sum_{k=1}^{|U_{ij}|} t_{ij}^k * w_k \tag{1}$$

where t_{ij} is the vector of temporal interaction trust of agent i in its partner j and w is the vector of weighted interaction.

It is easy to construct the vector of weighted interaction based on Regular Decreasing Monotone (RDM) linguistic quantifier function, which is given in the following definition.

Definition 3. *([9]) The function $q : [0,1] \to [0,1]$ is a RDM one, if and only if it satisfies the following conditions:*

$$\begin{aligned} (i)\ & q(0) = 1 \\ (ii)\ & q(1) = 0 \\ (iii)\ & q(i_1) \geqslant q(i_2) \text{ if } i_1 < i_2. \end{aligned}$$

Proposition 1. *([9]) Suppose that q is a RDM function. Then the vector $w = (w_i)$ generated by the following formula*

$$w_i = q\left(\frac{i-1}{| U_{ij} |}\right) - q\left(\frac{i}{| U_{ij} |}\right) \text{ for } i = 1, ..., | U_{ij} |$$

is the vector of weighted interaction.

3.2 Class of Reputation Trust Functions

In this section, we denote $\mathcal{W}_i \subseteq \mathcal{A}$ to be a world of agents, with whom agent i has interacted in the past. Intuitively, when an agent i needs to be based on other agents in $\mathcal{W}_i$ for evaluating an agent j (j is not in $\mathcal{W}_i$), its trust value on j must be greater than minimizing and smaller maximizing all values of partner values.

Definition 4. *The function $t : \mathcal{A} \times \mathcal{A} \rightarrow [0,1]$ is a reputation one in the world $\mathcal{W}_i$ iff it satisfies the following conditions:*

$$min_{x \in \mathcal{W}_i}\ t(x,j) \leq t(i,j) \leq max_{x \in \mathcal{W}_i}\ t(x,j)$$

We have the following proposition.

Proposition 2. *Suppose that $\mathcal{W}_i$ is a world composed of agents, with whom the agent i has interacted. The function t_R computed by means of average of individual trusts*

$$t_R(i,j) = \frac{1}{\mid \mathcal{W}_i \mid} \sum_{l \in \mathcal{W}_i} t(l,j)$$

is the reputation function.

3.3 Class of Similarity Trust Functions

In the reality, a truster i may infer a trust value on a trustee j by reasoning itself on the similarity of the trustee with other well known trustees. There are several methods to measure the similarity between two objects (e.g., [10]). Without loss of generality, assume that $\{a^1, a^2, ...a^n\}$ are attributes that are used to measure the similarity between two agents. In order to keep our model as simple as possible, we use *distance* between two agents based on a weighted average operator over the differences of all considered attributes of two agents.

Definition 5. *Suppose that a_i^k and a_j^k are values on the attribute a^k, after having normalized, of agent i and agent j, respectively. The difference between agent i and agent j ($i, j \in \mathcal{A}$) on attribute a^k is defined as follows:*

$$d_{ij}^k = \mid a_i^k - a_j^k \mid \quad (2)$$

The trust estimation of truster i on trustee j via another trustee l is based on the combination of the experience trust of i about l and the similarity between l and j. Intuitively, this computation must satisfy the following conditions:

- The more the experience trust of i about l is high, the more the similar trust is high;
- The more the difference between l and j is low, the more the trust value is close to the experience trust of the well known trustee.

These constraints may be represented by the following *Similar-based Trust Function - STF*:

Definition 6. *A function* $t_{sim} : [0,1]^2 \rightarrow [0,1]$ *is called the similar trust function, denote STF, if and only if it satisfies the following conditions:*

$$(i).\ t_{sim}(e_1, d) \leqslant t_{sim}(e_2, d) \ \text{if}\ e_1 \leqslant e_2;$$
$$(ii).\ \mid e - t_{sim}(e, d_1) \mid \leqslant \mid e - t_{sim}(e, d_2) \mid \ \text{if}\ d_1 \geqslant d_2;$$

The individual similar trust of a truster i on a trustee j via the similarity between the trustee j and another trustee l is defined as follows:

Definition 7. *The individual similar trust of a truster* i *about a trustee* j *via the similarity between the trustee* j *and another trustee* l *is a mapping* $f_s : [0,1] \times [0,1] \rightarrow [0,1]$ *from the experience trust of truster* i *about trustee* l *and the difference between trustee* l *and trustee* j*:*

$$s_{ij}^{l} = f_s(E_{il}, d_{lj}) \tag{3}$$

where f_s *is a STF function,* E_{il} *is the experience trust of truster* i *on trustee* l*,* d_{lj} *is the difference between agent* j *and agent* l*.*

3.4 Class of Functions against Liars

Suppose that $X_{ij} \subseteq A$ is a set of agents that agent i refers their trust from agent j and that agent i has already at least one transaction with them. We assume that an agent always trusts on itself. So, the trust of referee j from the point of view of agent i is determined based on the difference between experience trust E_{ik} and the trust r_{ik}^{j} of agent i in agent k referred via agent j (for all $k \in X_{ij}$). Intuitively, the trust of referee must satisfy these following conditions:

- The more the difference between E_{ik} and r_{ik}^{j} is big, the less agent i trust on the referee j, and conversely;
- The more the difference between E_{ik} and r_{ik}^{j} is small, the more agent i trusts on the referee j.

These constraints are defined by the following *liar-against function*:

Definition 8. *A function* $h : [0,1] \times [0,1] \rightarrow [0,1]$ *is called the liar-against function iff it satisfies the following conditions:*

$$h(e_1, r_1) \leqslant h(e_2, r_2) \ \text{if}\ \mid e_1 - r_1 \mid \geqslant \mid e_2 - r_2 \mid$$

Definition 9. *Referee trust* $S_{il} : \mathcal{A} \times \mathcal{A} \rightarrow [0,1]$ *of agent* i *on the referee* l *is defined by the formula:*

$$S_{il}(i,j) = \frac{1}{\mid X_{il} \mid} * \sum_{j \in X_{il}} h(E_{ij}, r_{ij}^{l}) \tag{4}$$

in which h *is a liar-against function and* E_{ij} *is the experience trust of* i *on* j*,* r_{ij}^{l} *is the reputation trust of agent* i *on agent* j *that is referred via referee* l*.*

Proposition 3. *A function $h : [0,1] \times [0,1] \rightarrow [0,1]$ defined by the formula*

$$h(x,y) = 1- \mid x - y \mid$$

is a liar-against function.

3.5 Class of Composition Functions

Intuitively, the combination trust of types of trusts must satisfy the following conditions:

- It must neither lower than the minimal and nor higher the maximal of experience trust and reference trust;
- The more the experience trust is high, the more the combination trust is high;
- The more the reference trust is high, the more the combination trust is high.

These constraints may be represented by the following combination function.

Definition 10. *A function $t_c : [0,1]^2 \rightarrow [0,1]$ is called the combination function iff it satisfies the following conditions:*

(i) $min(x_1, x_2) \leqslant t(x_1, x_2) \leqslant max(x_1, x_2)$;
(ii) $t(x_1, x_2) \leqslant t(y_1, y_2)$ *for each* $i = 1, 2$ *such that* $x_i \leqslant y_i$.

Proposition 4. *Suppose that $f_1, \cdots, f_n$ are trust functions. The function f defined by the formula*

$$f(i,j) = w_1 * f_1(i,j) + w_2 * f_2(i,j) + \cdots + w_n * f_n(i,j)$$

in which $w = (w_1, \ldots, w_n)$ is a weighted vector, $w_1 + \cdots + w_n = 1$, is a combination function.

Definition 11. *The combination trust is defined by the following formula*

$$f(i,j) = w_1 * f_1(i,j) + w_2 * f_2(i,j) + w_3 * f_3(i,j) + w_4 * f_4(i,j)$$

in which $w = (w_1, \cdots, w_4)$ is a weighted vector, $w_1 + w_2 + w_3 + w_4 = 1$ and f_1, f_2, f_3, f_4 are experience trust, reputation trust, similarity trust, liar-against functions, respectively.

4 Conclusion & Discussion

In this paper, we have introduced a formalization of concepts on types of trust. Based on these definition, we have described classes of trust functions, which contribute to computational trust models for estimating trust values. These functional classes represent features of the corresponding types of trust. We are currently considering an implementation and evaluation of function classes as well as their applications. The research results will be presented in our future work.

References

[1] J. Carter, E. Bitting, and A. Ghorbani, *Reputation formalization for an information-sharing multi-agent sytem*, Computational Intelligence, 18(2002), 515-534.

[2] Y. Mass and O. Shehory, *Distributed trust in open multi-agent systems*, In R. Falcone, M. Singh, and Y. H. Tan, editors, Trust in Cyber-societies,Springer-Verlag, Berlin (2001), 159-173.

[3] V.Mareeswari, Dr. E. Sathiyamoorthy, *A Survey on Trust in Semantic Web Services, International Journal of Scientific & Engineering Research*, 3(2012), 1-5.

[4] Jordi Sabater and Carles Sierra, *Review on Computational Trust and Reputation Models*, Artificial Intelligence Review 24(2005), 33-60.

[5] T. Grandison and H. Reichgelt, *Formalizing trust-based decision making in electronic commerce transactions*, Proceedings of the First International Workshop on Context Awareness and Trust Canada (2007), 13-28.

[6] T. D. Huynh, N. R. Jennings, and N. R. Shadbolt, *An integrated trust and reputation model for open multi-agent systems*, Journal of Autonomous Agents and Multi-Agent Systems, 13(2006), 119-154.

[7] M. H. Nguyen and D. Q. Tran, *A computational trust model with trustworthiness against liars in multiagent systems*, Proceedings of The 4th International Conference on Computational Collective Intelligence Technologies and Applications, Springer-Verlag Berlin Heidelberg (2012), 446-455.

[8] M. H. Nguyen and D. Q. Tran, *A multi-issue trust model in multiagent systems: A mathematical approach*, South-East Asian Journal of Sciences, 1 (2012), 46–56.

[9] M. H. Nguyen and D. Q. Tran, *A combination trust model for mutiagent systems*, International Journal of Innovative Computing, Information and Control (2013), to appear.

[10] Dekang Lin, *An Information-Theoretic Definition of Similarity*, In Proc. 15th International Conf. on Machine Learning, pages 296304.Morgan Kaufmann, San Francisco, CA, 1998.

[11] S. D. Ramchurn, C. Sierra, L. Godo, and N. R. Jennings, *Devising a trust model for multi-agent interactions using confidence and reputation*, International Journal of Applied Artificial Intelligence, 18(2004), 833-852.

[12] D. Q. Tran and M. H. Nguyen, *Modeling trust in open distributed multiagent systems*, East-West Journal of Mathematics, Special issue for Contribution in Mathematics and Applications III (2010), 98–108.

[13] P. Victor, C. Cornelis, M. De Cock, and P. Pinheiro da Silva, *Gradual trust and distrust in recommender systems*, Journal of Fuzzy Sets and Systems, 160 (2009), 1367-1382.