Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Environmental Sciences 11 (2011) 452 – 459

Procedia Environmental Sciences

A Trust Model Based on Cloud Model and Bayesian Networks

Bo Jin[a],Yong Wang[b], Zhenyan Liu[b], Jingfeng Xue[b]

[a]*Key Lab of Information Network Security of Ministry of Public Security(The Third Research Institute of Ministry of Public Security),Shanghai, China,jinbo@stars.org.cn*

[b]*School of Software,Beijing Institute of Technology Beijing, China,wangyong@bit.edu.cn*

Abstract

the Internet has been becoming the most important infrastructure for distributed applications which are composed of online services. In such open and dynamic environment, service selection becomes a challenge. The approaches based on subjective trust models are more adaptive and efficient than traditional binary logic based approaches. Most well known trust models use probability or fuzzy set theory to hold randomness or fuzziness respectively. Only cloud model based models consider both aspects of uncertainty. Although cloud model is ideal for representing trust degrees, it is not efficient for context aware trust evaluation and dynamic updates. By contrast, Bayesian network as an uncertain reasoning tool is more efficient for dynamic trust evaluation. An uncertain trust model that combines cloud model and Bayesian network is proposed in this paper.

*Keywords:*trust model; cloud model;Bayesian network; context aware; unceratinty

1.Introduction

In recent years, the popularity of the Internet boosts many new concepts such as cloud computing, “The Internet of Things”, and Internetware. The common idea of these concepts is service-oriented, which means that the distributed applications are constructed based on independent component services with standard interfaces [1]. How to select trustworthy services automatically becomes the key part of software development. Traditional security approaches such as authentication is binary logic, which cannot present multiple trust degrees or the human’s service selection principle of good enough. The subjective trust models can make up for the deficiencies by reasoning about a service’s trustworthiness in future interactions according to one entity’s direct interactions with that entity and recommendations (ratings) from other entities.

Trust is a concept with many uncertainties, among which, randomness and fuzziness are the two most important uncertainties. To grasp the uncertainty of trust accurately, most well known trust models use

doi:10.1016/j.proenv.2011.12.072

probability or fuzzy set theory to hold randomness or fuzziness respectively. Although the randomness and fuzziness are quite different in nature, many linguistic concepts contain simultaneous randomness and fuzziness. Keeping this in mind, Li et al. proposed a new cognitive model- Cloud model [2], which can synthetically describe the randomness and fuzziness of concepts and implement the uncertain transformation between a qualitative concept and its quantitative instantiations. During recent years, several researchers proposed cloud model based trust models in order to consider both randomness and fuzziness in trust evaluation.

Although cloud model is ideal for representing uncertain trust value, its logical operations and algebra operations lack sound theoretical basis. As the result, all existing trust models [3], [4], [5] based on cloud model have two common shortcomings in the aspect of trust aggregation (i.e. an agent usually estimates an unknown service entity's trust value by aggregating recommendations from other reliable agents.). First, the trust aggregation relies on cloud algebra operations, which is in fact the weighted average of cloud numerical parameters, and all weights are preset according to the trustor's experience, which cannot reflect the difference between recommendations and the ratings given by the source trustor, and causes to reduce the aggregation accuracy. Second, they cannot resist malicious recommendation attacks, because the reliability of recommendations is not evaluated and all recommendations are treated equally. In addition, these models don't consider context information, which makes it impossible to evaluate trust values and make decisions according to context information.

Our previous research work shows that Bayesian networks can help make context aware trust evaluation and aggregation in a rational (sound theoretical basis), intuitive (graphical representation) and robust (malicious recommendation attacks resistant) way [6]. So, we propose a trust model to combine cloud model and Bayesian networks, which can represent and evaluate the uncertainty of trust more accurately and efficiently.

2.Trust representation using cloud model

2.1.Cloud Model

Given a qualitative concept T defined over a universe of discourse U, let $x \in U$ is a random instantiation of the concept T and $\mu T(x) \in [0,1]$ is the certainty degree of x belonging to T ,which corresponds to a random number with a steady tendency. Then, the distribution of x in the universe U can be defined as a cloud and x can be called as a cloud drop.

A cloud describes the overall quantitative property of a concept by the three numerical characteristics as follows:

- Expectation *Ex* is the mathematical expectation of the cloud drops belonging to a concept in the universal.
- Entropy *En* represents the uncertainty measurement of a qualitative concept. It is determined by both the randomness and the fuzziness of the concept. In one aspect, as the measurement of randomness, *En* reflects the dispersing extent of the cloud drops and in the other aspect, it is also the measurement of fuzziness, representing the scope of the universe that can be accepted by the concept.
- Hyperentropy *He* is the uncertain degree of entropy *En*.

2.2.Trust Cloud

We use cloud to represent subjective trust, called trust cloud. The universe of discourse U= [0, n], n is any positive integer. Trust T is a qualitative concept defined over U. Because trust for a service entity is

evaluated from ratings from raters or recommenders, any rating r∈U can be regarded as a random instantiation of T. Every r is a cloud drop of trust cloud, which means a quantitative instantiation of the qualitative concept T. The certainty degree of r belonging to T is denoted by μ T(r)∈[0, 1].

Besides of cloud drops, we can also describe the overall quantitative property of T by Ex, En, and He. That is to say, the overall trust for a service entity can be represented using a tuple T (Ex, En, He). Next, we will describe briefly how to compute Ex, En, He, and μ T(r) from all r.

In real rating systems, it's common to rate service quality using discrete satisfactory levels (level1,…,leveln). Each level represents the extent to which an agent is satisfied with the interaction, in which level1 means "extremely unsatisfied" and leveln means "extremely satisfied". A rating r can be any integer in set I= {1, 2,⋯, n}, obviously, $I \subset U$.

We use a Bayesian network to calculate Ex, which takes all cloud drops r together with the context information as evidence, and the expectation of the cloud drops is Ex. Ex changes when new rating r is taken, and the newer the rating, the closer it is to the real value of Ex. So we adopt a time decay mechanism to make newer ratings' effect on Ex stronger. Please refer to Part III for the details of Ex calculation.

As to entropy En calculation, Li et.al use the standard deviation or the first-order absolute central moment of all cloud drops [7], we improve the algorithm by considering time decay in Ex calculation, as in

$$En = \frac{1}{i}\sum_{j=1}^{i}\left|r_j - Ex_i\right|, \quad i \geq 2 \cdot \tag{1}$$

Hyperentropy *He* is calculated as the first-order absolute central moment of all *En*, as in

$$He = \frac{1}{i}\sum_{j=1}^{i}\left|En_j - \frac{1}{i}\sum_{k=1}^{i} En_k\right|, \quad i \geq 3 \cdot \tag{2}$$

The certainty degree of r_i belonging to concept *T* is calculated using (3), in which *Ex* and *En* is the current value of expectation and entropy respectively.

$$\mu_T(r_i) = e^{-\frac{(r_i - Ex)^2}{2En^2}} . \tag{3}$$

3.Trust evaluation using bayesian networks

Subject trust is a context-specific concept, but neither existing trust models based on cloud model consider context information explicitly. The main reason is that it is not easy to integrate context information into cloud algebra and logic operations, which all take the overall three numerical characteristics as operands. If considering complex compound context information, the case would be even worse.

Our previous work [6] shows that Bayesian networks can be used to integrate context information into trust evaluation to improve accuracy. In this section, we will describe how to combine the trust cloud described in previous section with Bayesian networks to form a context-aware uncertain trust model.

3.1.Basic Context-aware Trust Evaluation

The rating about an interaction between agents can be one of n discrete levels (level1,…,leveln). Each level represents the extent to which an agent is satisfied with the interaction, in which level1 means "extremely unsatisfied" and leveln means "extremely satisfied". Context information of interactions should also be considered to improve trust evaluation accuracy. We consider m types of context information, and use $C_{ij_i}\ (i \in \{1,2,\cdots,m\})$ to represent the jth value of context type i, then the context information of an interaction can be represented as a tuple $C^{(C_{1j_1},C_{2j_2},\cdots,C_{mj_m})}$. It is reasonable to assume that states of different context types are independent. To sum up, a rating consists of a service level integer and a context tuple.

We use a naïve-Bayesian network for trust evaluation, where trust value is the root node and context information corresponds to leaves. Thus structure is highly extensible, when adding a new context type, the only thing to do is inserting a leaf node and existing conditional probability tables (CPTs) are still valid. Similarly, removing a context type doesn't have any effect to left nodes either. An example is shown in Fig.1, where trust is related to two types of contexts (Context_1 and Context_2). Node "Trust" has five states, from level1 to level5, corresponding to five possible ratings. Node "Context_1" has two states: context11 and context12, and node "Context_2"'s two states are context21 and context22 respectively. The CPTs (i.e. $P(Trust = level_k),\ k \in \{1,2,3,4,5\}$,

$$P(Context\_1 = context_{1j} \mid Trust = level_k),\ j \in \{1,2\}, k \in \{1,2,3,4,5\}$$

$$P(Context\_2 = context_{2j} \mid Trust = level_k),\ j \in \{1,2\}, k \in \{1,2,3,4,5\})$$

can be learned from the ratings (cases).

The time decay process can be done after a period of time or after learning some number of cases by fading old probabilities before taking new ratings into consideration. Equation (4) shows the CPT updating process of node "Trust", in which, P(m) (m ≥0) is the probability after m fading rounds; λ∈[0,1] is the fading factor.

$$P_{(0)}(Trust = level_k) = \frac{1}{n}$$

$$P_{(m+1)}(Trust = level_k) = \frac{kP_{(m)}(Trust = level_k)\cdot(1-\lambda)+\lambda}{m+(2-m)\lambda} \tag{4}$$

$$\lambda = e^{-En} \in [0,1]$$

We believe that the fading factor should reflect the stability of service entities' behavior, rather than taking fixed value as in most existing trust models. In fact, the certainty degree of ratings can represent the change of trust value, because the more the entity changes its behavior, the more the difference between the current rating and the expectation of trust value is, i.e., the smaller the certainty degree is. So we set the fading factor be the certainty degree of the last (newest) rating that has been taken by the Bayesian network as evidence.

Given the ratings, the trust evaluation process, that is the calculation of numerical characteristics (i.e., Ex, En, He) and each rating's certainty degree (i.e., μ T(r)), can be described as Algorithm 1.

Algorithm 1 Basic context-aware trust evaluation

Input: The set of ratings (cloud drops) *R* which includes context information

Output: Trust cloud's three parameters *Ex*, *En*, *He* and each rating's certainty degree $\mu_T(r)$

Step1: Initialize all the CPTs to be uniform distribution

Step2: $i = 1$

repeat

Read in a rating r_i and related context information tuple *C* from *R*

If needed, do time decay process as in (4)

Use the rating as a new case to update the CPTs

$i = i + 1$

until read in all ratings

Step3: Infer the probability that an agent's service quality is on $level_k$ in each different context C (i.e., $P(Trust = level_k | C)$, $k \in \{1, 2, \cdots, n\}$)

Step4: Calculate Ex in each different context C as the expectation of node "Trust" in context C using

$$Ex = \sum_{k=1}^{n} P(Trust = level_k | C) \times k$$

Step5: Calculate En in each different context C as in (1)

Step6: Calculate He in each different context C as in (2)

Step7:

for j = 1 to i -1 **do**

Calculate $\mu_T(r_j)$ in the related context C as in (3)

end for

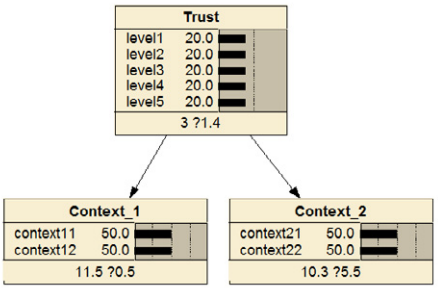

Figure 1. Bayesian network for context dependent trust estimation.

3.2.Unfair Rating Filtering

In order to avoid unfair ratings affecting the accuracy of trust evaluation, trustors should first evaluate raters' reliability and then select the most reliable ones. For each rater, its reliability can be estimated by the average certainty degree (μ T(r)) of all ratings from it. The larger the average certainty degree, the more reliable is the rater. For sake of computation, we select at most two reliable raters whose reliability is larger than some specific value, say 0.6, and filter out all other ones.

3.3.Service Provider Selection Based on Trust Cloud

We argue that uncertainty of trust should be taken into consideration for service provider selection. So, we use trust cloud as the decision criteria, that is to say, not only the trust value (which is represented by Ex) but also its uncertainty (which can be represented by En＋He according to (1) and (2)) is compared. Obviously, entities with higher trust value and lower uncertainty are more trustworthy, while those with lower trust value and higher uncertainty are not trustworthy.

Various service entity selection approaches can be designed for specific applications. In this paper, we select the entity with smaller En＋He from the two entities with the most largest Ex.

4.Experimental analysis

4.1.Experiment Setting

There are 100 entities, 60 of which belong to group A and 40 belong to group B. Each entity's behavior is determined by a behavioral probability tuple (p1, p2, p3), which is corresponding to service level (level1, level2, level3). We consider two context attributes and both have two states as described in previous section, so there are four different context situations. The original (change later) probability tuples for the two groups are shown in Table I.

The simulation is divided into 100 rounds and around 25 rounds for each situation. In each round, each entity selects a service provider entity according to some criteria (see Section B). In order to simulate the dynamic nature of entity behavior, the probability tuples are changed randomly after every 10 rounds as in (6).

$$\begin{cases} p_1 = p_1 - \Delta/2, p_2 = p_2 + \Delta, p_3 = p_3 - \Delta/2 & \text{(with probability 0.33)} \\ p_1 = p_1 + \Delta/2, p_2 = p_2 - \Delta, p_3 = p_3 + \Delta/2 & \text{(with probability 0.33)} \\ p_1 = p_1, p_2 = p_2, p_3 = p_3 & \text{(with probability 0.34)} \\ \Delta = 0.01 \end{cases} \tag{6}$$

So the entities fade all the CPTs each time after learning every 10 cases, and then incorporate the latest cases. The fading factor λ equals 0.01. The proportion of unfair raters is 40%.

Table 1. Entities' original behavior patterns

Behavioral Probability Tuple	**Group**	
	A	***B***
$Context_{11}$, $Context_{21}$	(0.05,0.05,0.9)	(0.9,0.05,0.05)
$Context_{11}$, $Context_{22}$	(0.1,0.1,0.8)	(0.8,0.1,0.1)
$Context_{12}$, $Context_{21}$	(0.1,0.2,0.7)	(0.7,0.2,0.1)
$Context_{12}$, $Context_{22}$	(0.2,0.2,0.6)	(0.6,0.2,0.2)

4.2.Experiment Results

We use successful service ratio to evaluate the accuracy of our trust model. If a selected service provider behave good, that is its service level is level3, then the service is successful, otherwise, it is fail. We compare the ratios in the following three different scenarios.

- Entities don't use trust evaluation, that is, they select service providers randomly.
- Entities use the trust evaluation but don't consider context information.
- Entities use the trust evaluation and consider context information.

Table II shows the results in four possible context situations.

Table II Successful service ratio in different scenarios

Successful Service Ratio (%)	**Trust Evaluation Scenarios**		
	Random	***Trust without context***	***Trust with context***
$Context_{11}$, $Context_{21}$	56.96	83.47	88.81

Successful Service Ratio (%)	Trust Evaluation Scenarios		
	Random	*Trust without context*	*Trust with context*
$Context_{11}$, $Context_{22}$	57.11	78.26	83.57
$Context_{12}$, $Context_{21}$	57.62	77.19	80.79
$Context_{12}$, $Context_{22}$	61.25	80.8	79.11

We can see from Table II that our trust model with can help improve the successful service ratios largely, from around 60% (Random) to higher than 77% (Trust without context). In addition, context information has good effect on estimation accuracy, the ratios in the right most column (Trust without context) are around 5% higher than those in the middle column (Trust with context). The only exception is in the last context situation (Context12, Context22), because entity behavior is more uncertain than in other situations, the ratio with context (79.11%) is a little bit lower than that without context (80.8%).

As entities only select reliable recommenders, it would be expected that the proportion of unfair recommenders has little effect on the accuracy of estimation. The proportion of unfair recommenders in the experiments shown by Table II is 40%. When the proportion of unfair recommenders is 70%, the smallest successful service ratio is as high as 75% (Trust without context).

5.Conclusions

A trust evaluation model which combines the cloud model and Bayesian networks is proposed in this paper. The model has the following features.

- The uncertainty of trust is explicitly represented and evaluated using sound theories for uncertain reasoning, which make the evaluation more accurate even when entities' behaviors change dynamically.
- Context information is explicitly integrated into trust evaluation to make the service provider selection be context-ware.

Our future works will focus on unfair rating filtering and trust aggregation.

Acknowledgment

This paper is supported by the Opening Project of Key Lab of Information Network Security of Ministry of Public Security (The Third Research Institute of Ministry of Public Security), China (Grant No. C10604); and by the Key Program of Shanghai Committee of Science and Technology, China (Grant No. 10511501502); and by the Soft Science Research Program of Shanghai Committee of Science and Technology, China (Grant No. 11692101800).

References

[1] W.T. Tsai, Y.N. Chen, X. Sun, C. Cheng, G. Bitter, and M. White, "Service-Oriented Computing," Learning & Leading with Technology, vol. 35, May 2008, pp. 28-30.

[2] D.Y. Li, C.Y. Liu, and W.Y. Gan, "A new cognitive model: Cloud model," International Journal of Intelligent Systems, vol. 24, March 2009, pp. 357-375.

[3] X.Y. Meng, G.W. Zhang, J.C. Kang, H.S. Li, and D.Y. Li, "A New Subjective Trust Model Based on Cloud Model," Proc. IEEE International Conference on Networking, Sensing and Control (ICNSC 2008) , IEEE Press, Apr. 2008, pp. 1125-1130.

[4] S.X. Wang, L. Zhang, S. Wang, and N. Ma, "An evaluation approach of subjective trust based on cloud model," Proc. 2008 International Conference on Computer Science and Software Engineering (CSSE 2008), IEEE Press, Dec. 2008, vol.3, pp.1062-1068.

[5] F. Lu, H.Z. Wu, "Research of Trust Valuation and Decision-making Based on Cloud Model in Grid Environment," Journal of System Simulation, vol. 21, Jan. 2009, pp.421-426.

[6] Y. Wang, M. Li, J.F. Xue, J.J. Hu, L.F. Zhang, and L.J. Liao, "A Context-aware Trust Establishment and Mapping Framework for Web Applications", Proc. 2007 International Conference on Computational Intelligence and Security (CIS'07), IEEE Press, Dec. 2007, pp. 892-896 doi:10.1109/CIS.2007.13.

[7] D. Y. Li, and Y. Du, Artificial Intelligence with Uncertainty. National Defense Industry Press, 2005.