Trust and Reputation Model Based on Causal Induction Method

[1]Lourembam Suraj Singh, [2]Heisnam Rohen Singh, [3]Arambam Neelima, [4]Sarangthem Ibotombi Singh

[1]Dept. of Computer Centre, Mizoram University, Mizoram, India

[2,3]Dept. of IT, Tezpur University, Assam, India

[4]Dept. of CSE, Tezpur University, Assam, India

Abstract

Trust is an important concept that acts as the building block for successful interaction of any social network. Without trust, it will be difficult to achieve efficient and comfortable interactions among the users. For this reason, trust and reputation have been incorporated in many web services. Nowadays, many web services have become user centric with independent interoperability. This trend requires that interpersonal trust should be examined and maintained. The purpose of this paper is to present an agent based model based on the causal induction method to evaluate the interpersonal trust among the users. In this model, we will try to evaluate how much 'A' trust 'B' based on their past experience and information from other cliques. One important property of trust is that users prefer recommendations from their trusted cliques than recommendation from unknown source. Thus the closeness of the relationship also plays an important role in shaping the decision of users. To our best knowledge, this work is first to apply causal induction method in building a trust model of online social networks.

Keywords

Trust, Reputation, Cause and Effect, Causal Induction, ΔP, Causal Strength, Causal Power

I. Introduction

Many Large scale distributed systems, such as peer to peer networks, web services and e-commerce consists of a large number of agents. The proper functioning of these systems depends on the efficient and reliable interactions among the agents which in turn depend on the interpersonal trust among the agents. Without trust it will be impossible to facilitate efficient and easy interaction among the users. Moreover there are large numbers of agents and are heterogeneous; some agents are kind and will offer good and reliable services while some are bad and will try to harm other agents or provide bad services. The traditional measures such as cryptographic security, authentication and authorization methods cannot handle these kinds of systems where proper functioning of the system is dependent on the evaluation of interpersonal trust. In such kind of scenario, measures of trust and reputation is used to solve the problem by allowing agents to distinguish good from bad agents and thus encouraging the agents to be trustworthy.

The rest of the paper is organized as follows: Section II, discusses the notion of trust and reputation. Section III, describes the causal induction method. Section IV, introduces our approach to developing a causal induction based trust and reputation model. The experimental design and the results are presented in section V, and VI. Finally, we make conclusions in section VII.

II. Trust and Reputation

The concept of trust applies to many areas and each area has specific meaning based on the context where it is used. But the basic definition of trust remains the same and it can be defined as—

" 'A' trust 'B', if 'A' takes up an action based on the assumption that B's future action will lead to a satisfiable outcome".

Thus, trust in the context of the agents can be defined as the confident expectations in the reliability, efficiency, or honesty in providing services by the agent. Reputation is what is generally said or believed about the agent by other agents. Agent A's trust about B resulted from A's past interaction experience with B. It indicates A's opinion towards B's competence in providing services. On the other hand, B's reputation is A's opinion towards B's competence in providing services which is obtained by consulting other agents. Depending on B's reputation, A may or may not interact with B. Once they interact, A can establish its own trust towards B. Taking an example, if A personally knows B, i.e. if A has already interacted with B, then A develops some kind of trust towards B. Thus A will have some idea that if it seeks services from B, whether B will respond to it or not. If A has not interacted with B, then A can take the help of C that has already interacted with B. Thus A's information about B that it receives from C forms the reputation of B. By using this reputation, A can either interact or ignore B. If A makes interaction with B, then A can develop its own trust towards B which will be used for future interactions.

III. Causal Induction Method

A. Causal Model

A causal model is an abstract model that uses cause and effect logic to describe the nature of a system focusing on causal factors. In causal model, events are observed as cause and effect. The main notion of causal model is that, cause precedes effect. It means that if a cause occurs it will lead to the occurrence of the effect or at least increases the chance of the occurrence of the effect. So, the main problem of causal learning is estimating the strength of a causal relationship or inferring that such a relationship exists.

B. Causal Graphical Model

Causal graphical model provides a formalism for learning and reasoning about causal relationships and it has become a current topic of research in computer science and statistics[9],[10]. In causal graphical model, the cause and effect logic is represented as graphs. In this model, all the relevant variables are represented as nodes. If a direct causal relationship exists i.e. if a particular cause causes an effect to occur then their corresponding nodes are joined by a directed link, or edges, from the cause to the effect. In this model, if a variable 'X' is changed then only its descendents, nodes directed away from 'X', are affected. For example, let us consider a case. When you reached home after work and enter your room, you find out that your room is wet. Then there are two possible causes—

- The sprayer was on or
- The water tape was broken or leaking

Here the states of the floor, sprayer and tape are the variables being considered, so each is assigned a node. The link between Sprayer and the Floor and the link between Tape and Floor is an indication that the states of Sprayer and Floor are the direct causes of the Floor getting wet. This can be represented in the graphical model as—

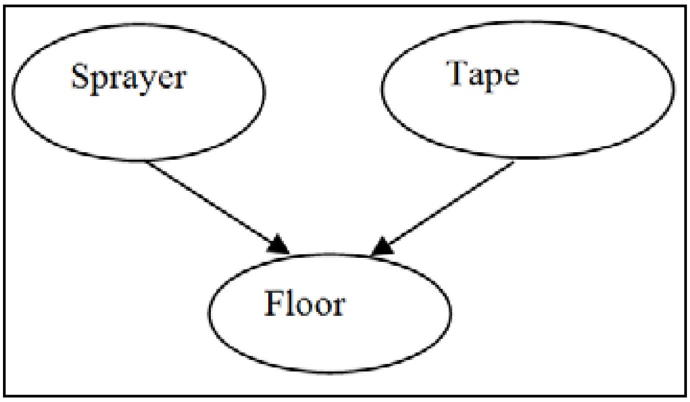


Fig. 1: Causal Graphical Model Showing the Relationship Between Floor Getting Wet and Occurrence of Sprayer or Tape

C. ∆P and Causal Power

∆P is the measure of the strength of the relationship between an effect and its possible cause. It was first suggested by [1] subsequently explored and applied in many studies [2-7]. ∆P can be calculated by taking the difference between the conditional probability of the target effect e given that the cause c has occurred, $P(e^+ \mid c^+)$, and the conditional probability of the effect given that c has not occurred, $P(e^+ \mid c^-)$. It can be expressed as—

$$\Delta P = P\ (e^+ \mid c^+)\ \text{-}\ P(e^+ \mid c^-)$$

$$= \frac{N(e^+,c^+)}{N(e^+,c^+) + N(e^-,c^+)} + \frac{N(e^+,c^-)}{N(e^+,c^-) + N(e^-,c^-)}$$

$N(e^+,c^+)$ is the number of cases in which effect occurs in the presence of cause.

$N(e^+,c^-)$ is the number of cases in which effect occurs in the absence of cause.

$N(e^-,c^+)$ is the number of cases in which cause is present but effect does not occur.

$N(e^-,c^-)$ is the number of cases in which both cause and effect is absent.

∆P thus can be expressed as the change in the probability of the effect occurring as a result of the occurrence of the cause.

Cheng [8] proposed Causal Power based on ∆P. Causal Power takes causal learning to be a problem of inferring the causal power of prospective causes [7]. The causal power can be expressed as—

$$Causal\ Power = \frac{\Delta P}{1 - P(e^+,c^-)}$$

Causal Power make a prediction that ∆P will have greater influence on the causal relationship when $P(e^+ \mid c^-)$ is large. Causal power can be considered as a case of e not occurring when c was not present and e will occur if c is introduced.

IV. Proposed Model

A. Scenario

We consider a network in which some peer acts as service providers i.e. servers and some peer acts as agents i.e. they use the service provided by servers. Each agent develops two kinds of trusts, the trust in the server's competence in providing services and the trust in other agent's reliability in providing recommendations. We made an assumption that all agents are honest in providing recommendations.

B. Working of the Model

We assume that the trust level of each of the server in providing the services can be divided into four levels—

- Very Low(VL)
- Low(L)
- Medium(M)
- High(H)

In order to apply the concept of causal induction method, we assume that all the states are nodes. Any server can be at any level, say 'X', at any time't'. Then it make transition to level 'Y' at time't+1'. In such a scenario, we assume that being at state 'X' causes the transition to state 'Y' or at least increases the probability of occurrence of state 'Y'. So, this relationship can be expressed graphically as—

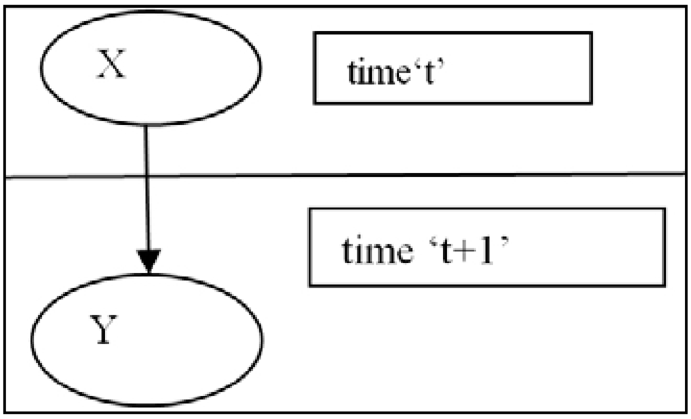


Fig. 2: Graphically Representing Transition from State 'X' to State 'Y'

Assume at time't' state of a server is 'L'. Now, at time't+1', the server can made transition to any of the four possible states i.e. VL, L, M and H. So, graphically it can be represented as—

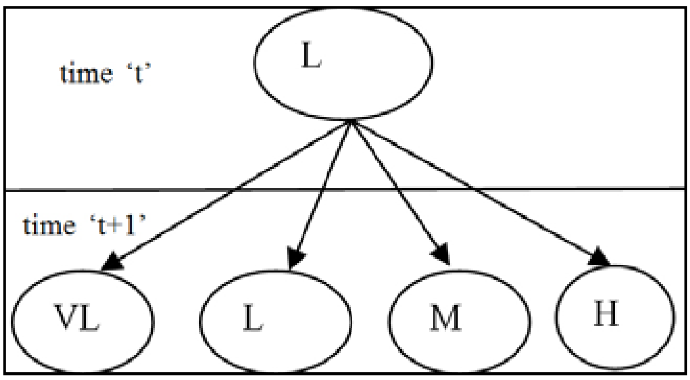


Fig. 3: Graphical Representation of Transition from State 'L' at Time 't' to All Possible States at Time 't+1'

Here we consider 'L' as the possible cause and 'VL', 'L','M', and 'H' as its effect. Now, extending the above concept to all the possible states, we can graphically represent the whole scenario as—

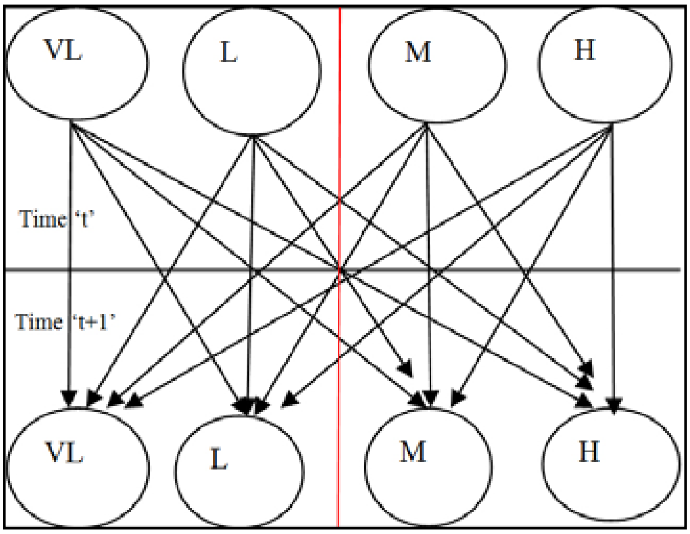


Fig. 4: Graphically Representing All Possible State Transition for a Given Server

Every agent also maintains a contingency table for each server. The contingency table of a given server is represented as a matrix of size 4x4 and is shown below—

Table 1: Contingency Table

n(VL→VL)	n(VL→L)	n(VL→M)	n(VL→H)
n(L→VL)	n(L→L)	n(L→M)	n(L→H)
n(M→VL)	n(M→L)	n(M→M)	n(M→H)
n(H→VL)	n(H→L)	n(M→L)	n(H→H)

Each entry in the contingency table indicates the number of cases a server made transition from a given state to a new state. For example, n(L→M) indicates the number of times a given server made transition from state 'L' to state 'M'. Given a current trust level at time 't', agents will like to know the possible trust level at time 't+1'. Suppose at time't', the trust level of the server is Low (L). The strength of the service provider going from low level to medium level at time't+1' can be expressed mathematically as—

$\Delta P = P(M^+|L^+)$ — $P(M^+|L^{})$

$= P(M^+|L^+) - \{P(M^+|VL^+)+P(M^+|M^+)+P(M^+|H^+)\}$

ΔP can be easily obtained from the contingency table of the corresponding server.

We can think of the transition from state 'X' at time 't' to state 'Y' at time 't+1' as an event {X→Y} such that {X,Y} Є {VL,L,M,H}. So, all the possible transitions can be represented as two sets G_1 and G_2 where G_1 and G_2 are given by—

Set G_1 = {(VL→VL),(VL→L),(L→VL), (L→L), (M→VL), (M→L), (H→VL), (H→L)}

={X→Y, such that X Є {VL,L,M,H} and Y Є {VL,L}}

Set G_2 = {(VL→M), (VL→H), (L→M), (L→H), (M→M), (M→H), (H→M), (H→H)}

={X→Y, such that X Є {VL,L,M,H} and Y Є {M,H}}

Every transition can be viewed as an event belonging to either G_1 or G_2. Thus, transition (L→VL) can be viewed as an event belonging to G1 while (VL→H) is an event belonging to G_2.

Since we are interested in high reliability only, we find out the causal strength of a server at any time't' making a transition at time 't+1' such that the event can be represented by a member of G_2. This causal strength is used to determine the reliability of the server in providing services at time 't+1'. Suppose a user wants to access the services from the servers. Before interacting it has to decide with which server it has to interact. If a user happens to select an unreliable server, who provides bad quality services, the user will waste time and effort. This situation may occur several times which may cause frustration to the user. Trust and Reputation system can be used to solve this problem. Once a user has received a list of servers from which it can obtain its services, it can arrange this list based on its trust values and it can select the highest trusted server for its services. If the agent has no information about the servers, it can ask other users for recommendations. Each user has a set of rules so that the recommendations from the most trusted source is given more wightage than the recommendation from other source. The user can even discard recommendations from the untrusted source or can give very less weightage to their recommendations based on the policy the user adopt to handle the recommendations.

If Agent 'i' wants to access services from the servers, 'i' has to first find out the last interacting state of each server 'j', before each interaction. Let us assume that the last interacting state of the server is 'X'. Agent 'i' then looks up in its transition table and calculates ΔP and Causal Power. If t_{ij} and t'_{ij} denotes the trust that agent 'i' has in the server 'j' for getting good services obtained from ΔP and causal power respectively, then t_{ij} and t'_{ij} can be calculated as—

Case a: If Agent 'i' has not interacted with server 'j':

In this case, $t_{ij}=0$ and $t'_{ij} = 0$.

Case b: If Agent 'i' has interacted with server 'j':

In this case t_{ij} and t'_{ij} is given as,

$$t_{ij} = \Delta P_{ij}/2 = \frac{\{P(M^+|X^+) - P(M^+|X^-)\} + \{P(H^+|X^+) - P(H^+|X^-)\}}{2} \quad (1)$$

And

$$t'_{ij} = CP_{ij} = \frac{\Delta P_{ij}/2}{1 - \{P(M^+|X^-) + P(H^+|X^-)\}/2} \quad (2)$$

Mathematically it can be represented as—

$$t_{ij} = \begin{cases} 0; if\ i\ has\ not\ interacted\ with\ j \\ \Delta P_{ij}/2\,; otherwise \end{cases} \quad (3)$$

And

$$t'_{ij} = \begin{cases} 0; if\ i\ has\ not\ interacted\ with\ j \\ CP_{ij}\ as\ obtained\ from\ (2); otherwise \end{cases} \quad (4)$$

Agent 'i' then sends reputation requests to all the other agents for server 'j'. On receiving the reputation request, each agent 'k', then finds out the last interacting state with server 'j' (say Y) and calculates t_{kj} and t'_{kj} as given by—

$$t_{kj} = \begin{cases} 0; if\ k\ has\ not\ interacted\ with\ j \\ \Delta P_{kj}/2\,; otherwise \end{cases} \quad (5)$$

$$t'_{kj} = \begin{cases} 0; if\ i\ has\ not\ interacted\ with\ j \\ CP_{kj}\ as\ obtained\ from\ (2); otherwise \end{cases} \quad (6)$$

Thus, total trust that agent 'i' has in the j^{th} server is given by—

$$T_{ij} = t_{ij} + \frac{\sum_{k=1}^{n} W_{ik} * t_{kj}}{\sum_{k=1}^{n} W_{ik}} \quad (7)$$

 ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)

$$T'_{ij} = t'_{ij} + \frac{\sum_{k=1}^{n} W_{ik} * t'_{kj}}{\sum_{k=1}^{n} W_{ik}} \quad (8)$$

Where t_{kj} is the trust that k^{th} agent has in the server 'j' (obtained by (5)) . t_{ij} is the trust that agent 'i' has in the j^{th} server (obtained by (3)). t'_{kj} is the trust that k^{th} agent has in the server 'j' (obtained by (6)) . t'_{ij} is the trust that agent 'i' has in the j^{th} server (obtained by (4)) and n is the total number of agents from whom agent 'i' has taken recommendations. W_{ik} is the weight that the agent 'i' assigns to the recommendations from k^{th} agent.

Thus agent 'i' calculates trust values for all j, i.e. for all servers. It then selects the server with the largest value of trust and interacts with it.

C. Evaluation of the Interaction

Prior to every interaction, each server knows a value say 'k'. Depending on the nature of the server, the server decides to return a value to the interacting agent. There are four types of servers—

1. Honest Server

This types of server always returns the true value to the interacting agent i.e. it always returns value 'k' to the agent.

2. Dishonest Server

This type of server always returns a false value to the interacting agent i.e. it always returns a value other than 'k' to the agent.

3. Random Server

This type of server always returns a random value to the interacting agent and sometimes it may even return the value 'k' to the agent.

4. Neutral Server

This type of server always returns a constant value 'c' to the interacting agent for every interaction.

After each interaction, interacting agent can look up for the actual value. Depending on the actual value and the received value from the server, the agent then determines the state of the server for the interaction. If 'p' is the value that agent received from the server, then the agent determines the state of the sever as—

Case (a)

When $|p-k|=0$.

The agent assigns the current state of the server as H.

Case (b)

When $0<=|p-k|<\theta_1$, where θ_1 is a user defined value.

The agent assigns the current state of the server as M.

Case (c)

When $\theta_1<=|p-k|<\theta_2$, where θ_2 is user defined value.

The agent assigns the current state of the server as L.

Case (d)

When $\theta_2<=|p-k|$.

The agent assigns the current state of the server as VL.

If 'X' was the last state and 'Y' is the current state of the interacting server, then the new transition is X→Y, which can be viewed as an event 'E'. We then determine, if the interaction is successful or not based on the event 'E' as—

Case (a)

When X→Y is an event 'E' that can be represented in the set G_1:

If E is an event such that E can be represented in G_1, then the interaction is unsuccessful.

Case (b)

When X→Y is an event 'E' that can be represented in G_2:

If E is an event such that E can be represented in G_2, then the interaction is successful.

Mathematically it can be summarized as—

$$Result\ of\ interaction = \begin{cases} unsuccessfull;\ if\ X \rightarrow Y\ is\ an\ event\ E \\ \quad such\ that\ E \in S_1 \\ successfull;\ if\ X \rightarrow Y\ is\ an\ event\ E \\ \quad such\ that\ E \in S_2 \end{cases}$$

Based on the outcome of the interaction, the agent then modifies its contingency table for the interacting server.

D. Handling other agents' recommendations

If the recommendation value sent by the agent 'k' is matching with the outcome of the interaction with server 'j', agent i then increases the weight of the agent k in its weight table. If agent i finds out that agent k was giving a false recommendation, then it decreases the weight of the agent k in its weight table. Mathematically it can be shown as—

$$W_{ik} = \begin{cases} W_{ik} + \alpha, if\ the\ recommendation\ is \\ \quad matching\ with\ the\ outcome \\ W_{ik} - \alpha, if\ the\ recommendation\ is\ not \\ \quad matching\ with\ the\ outcome \end{cases}$$

α is a user defined value, and it controls how an agent shifts its behavior towards other agents. However, the weight is constrained to lie between 0 and 1 by considering the following cases—

Case a: When $W_{ik} >1$, then set $W_{ik} = 1$.

Case b: When $W_{ik} <0$, then set $W_{ik} = 0$.

V. Experiment

We set up an environment in which there are 50 agents and 10 servers. For simplicity we assume that all the servers can provide all the requirements of each agent. At the beginning, each agent knows few agents and few servers i.e. each agent has interacted with few agents and few servers. At the beginning of each simulation we initialize the weight of each agent. If an agent 'j' has made interactions with the agent 'k', then we initialize W_{ik} =1 otherwise we initialize W_{ik}= 0.5. We run the simulations for 100 numbers of interactions and each configuration is run for 10 times. After getting the result, we plot the graph taking the means of each result. We call the model based on ΔP as Causal Strength Model and the Model based on Causal Power as Causal Power Model.

The aim of the first experiment is to see if trust and reputation models based on the ΔP and Causal Power helps in agents selecting the server with better reliability. Thus, we design three models—

- Trust and reputation model based on ΔP.
- Trust and reputation model based on Causal Power and
- A simple model that interacts with a randomly selected server.

After getting the results, we plot the graph of the models with percentage of successful interactions against the number of interactions.

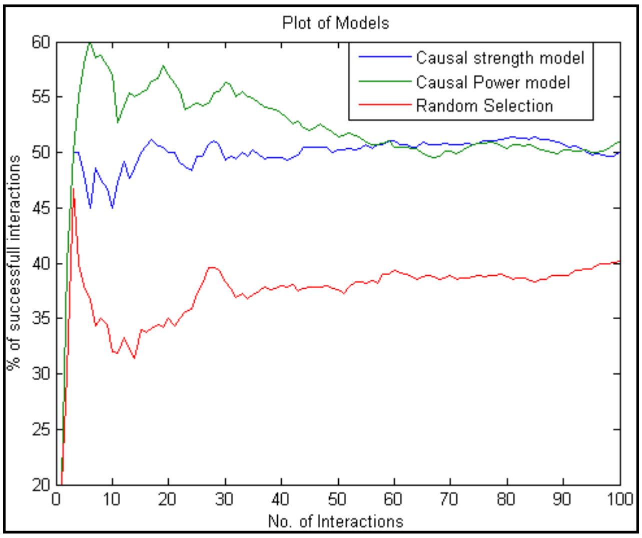


Fig. 5: Trust and Reputation Systems with Causal Induction Method v/s a Model with Random Selection

Fig. 5 shows that the trust and reputation model based on ΔP and Causal Power performs better than the model with no specific rule to determine which server to interact.

The aim of the second experiment is to see if the reputation model i.e. taking the views of the other agents helps in getting better performance in selecting reliable servers. For this setup, we compare four models—

- Trust and reputation model based on ΔP
- Trust model based on the ΔP without taking recommendations from other agents.
- Trust and reputation model based on the Causal Support method.
- Trust model based on the Causal Support method without taking recommendations from other agents.

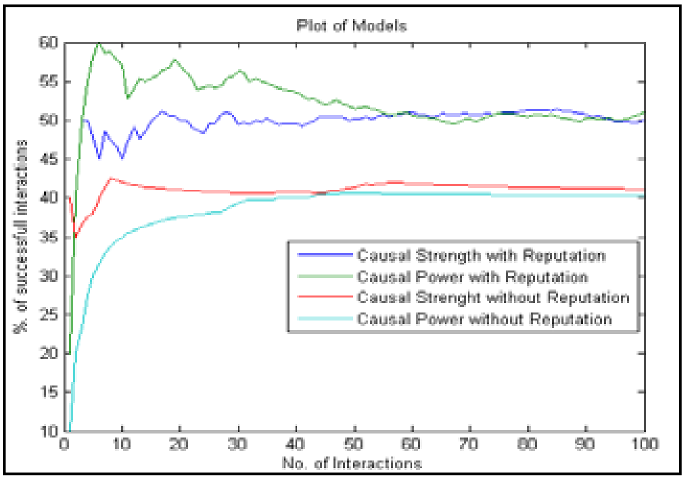


Fig. 6: Comparison of Model with Recommendation Evaluations v/s Model Without Recommendation Evaluation v/s a Model that Selects Servers Randomly

Fig. 6 shows that the model which considers recommendations from other agents performs better than the model that does not consider the evaluations from other agents.

VI. Conclusions

In this paper, we presented trust and reputation systems based on the ΔP and Causal Power. The result of the experiment shows that ΔP and Causal Power can be used to determine the trust value of the agents. From the above observations we can conclude that Causal Power gives slightly better result than ΔP. It also shows that system that considers recommendations from other agents performs better than the system that does not consider recommendations from other agents.

References

[1] H.M. Jenkins, W.C. Ward,"Judgment of contingency between responses and outcomes", Psychological Monographs, 79, 1965.

[2] L. G. Allan,"A note on measurement of contingency between two binary variables in judgment tasks", Bulletin of the Psychonomic Society, 15, 1980, pp. 147-149.

[3] L. G. Allan,"Human contingency judgments: Rule based or associative?", Psychological Bulletin, 114, 1993, pp. 435-448.

[4] L. G. Allan, H.M. Jenkins,"The effect of representations of binary variables on judgment of influence", Learning and Motivation 14, 1983, pp. 381-405.

[5] P.W. Cheng, K.J Holyoak,"Complex adaptive systems as intuitive statisticians: causality, contingency, and prediction".

[6] H.L Roitblat, J.-A Meyer (Edns),"Comparative approaches to Cognitive Science", Cambridge, MA:MIT Press, 1995, pp. 271-302.

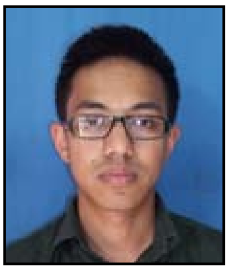
[7] E.R. Melz, P.W. Cheng, K.J. Holyoak, M.R. Waldmann, "Cue competition in humane categorization: Contingency or the Rescorla-Wagner rule, comments on Shanks", Journal of Experimental Psychology: Learning, Memory and Cognition, 19, 1991, pp. 1398-1410.

[8] T.L. Griffiths, J.B. Tenenbaum,"Structure and strength in causal induction", Cognitive Psychology 51, 2005, pp. 334-384.

[9] P. Cheng,"From co variation to causation", A causal power theory", Psychological Review 104, 1997, pp. 367-405.

[10] P. Spirtes, C. Glymour, R. Schienes,"Causation prediction and search", Springer-Verlag, New York, 1993.

[11] Y. Wang, J. Vassileva,"Bayesian Network-Based Trust Model", Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence (WI 2003), Halifax, Canada, 13-16 October 2003, pp. 372-378.



Lourembam Suraj Singh received his B.Tech in Computer Science and Engineering in 2009 from North Eastern Regional Institute of Science and Technology (NERIST), Itanagar, Arunachal Pradesh, India and M.tech in Information Technology in 2012 from Tezpur University, Assam, India. Currently, he is working as System Analyst at Mizoram University, Mizoram, India.

 ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)

Heisnam Rohen Singh received his B.Tech in Computer Science and Engineering in 2009 from Maharashtra Institute of Technology, Maharashtra, India and M.tech in Information Technology in 2012 from Tezpur University, Assam, India.

Arambam Neelima has received her B.Tech in Information Technology in 2010 from Shillong Engineering and Management College, Jorabat, India and M.tech in Information Technology in 2012 from Tezpur University, Assam, India.

S. Ibotombi Singh received the B.Sc. degree in Physics and the MCA degree from Manipur University in 1989 and 2004 respectively. Currently he is working as Assistant Professor in the Department of Computer Science & Engineering under the School of engineering, Tezpur University. He also served at the University of Delhi. He has published many papers in the areas of Data Mining, Trust and Reputation Systems in International Journals and Conferences. His current research interests are in Web services, trust management, network security, and GIS.