A Translation Mechanism for Recommendations

Pierpaolo Dondio, Luca Longo
dondiop@cs.tcd.ie, longol@cs.tcd.ie

School of Computer Science and Statistics,
Distributed System Group,

Trinity College Dublin, College Green 2, Dublin

June, 18th - 20th

IFIPTM 2008
Joint iTrust and PST Conferences on Privacy,
Trust Management and Security
Trondheim, Norway.

An important class of distributed Trust-based solutions is based on information sharing. This class encompasses Recommendation, Reputation systems and Social Networks

A basic requirement of such systems is the ability of participating agents to effectively communicate with others, receiving and sending messages that can be correctly interpreted.

Unfortunately, in open systems, it is not possible to postulate a common agreement about the representation of a rating, its semantic meaning, the cognitive and computational mechanisms behind a trust-rating formation.

If agents cannot understand each other, the feasibility of these trust solutions is fatally undetermined, even when all agents are acting honestly.

Social Scientists: Trust is not completely transferable

Social scientists agree to consider unqualified trust values not transferable and computational trust researchers are aware of the not completely transitivity of trust, proposing conditions and uncertain models.

Our Pragmatic approach: Trust judgements are worth being transferred

A more pragmatic approach would conclude that *qualified trust judgements are worth being transferred* as far as decisions taken considering others' opinions are better than the ones taken in isolation.

Defining the problem

Trust System 1
- Trust Value
- Trust Metric
- Trust Model
- Outcome Evaluation function
- Decision Making process

Domain Perception
- Domain Ontology
- Evidences DB

Trust System 1

Trust value *Component*$_1$ by XYZ: 0, 85 (*real*[0, 1])

A formula quantifying and aggregating concepts described in the model

Trust is a function of the stability of a software, its author's trustworthiness and recommendation

A good sw component has comments' clarity, simplicity & low number of bugs after 1 year

Threshold = 0.8, a set of policies/rules

Trust System 2

Trust value *Component*$_1$ by XYZ: true (boolean)

A formula quantifying and aggregating concepts described in the model

Trust is a function of its author's trustworthiness and the programming language used

A good sw component is portable and easy to read and modify by other software developer

Threshold = true, a set of policies/rules

Domain Perception 1

An ontology to describe facts of sw components

Sw *Component*$_1$ by XYZ crashed twice yesterday

Sw *Component*$_1$ by XYZ has been modified 10 times in the last 3 weeks

Domain Perception 2

An ontology to describe facts of sw components

Sw *Component*$_1$ by XYZ is written in Visual Basic

Sw *Component*$_1$ by XYZ is not working with Macintosh OS/X

Trust System 2
- Trust Value
- Trust Metric
- Trust Model
- Outcome Evaluation function
- Decision Making process

Domain Perception
- Domain Ontology
- Evidences DB

Two parties disclose some information in order to understand the degree of compatibility of their trust systems

Agents have been designed to interact with each other in a domain. We assume that their domain representations partially overlap. Differences can arise in their preferences, trust models or their past experiences, but we postulate that they are partially able to communicate (the domain ontology partially matches).

IFIPTM 2008

Strategies

1. the sharing of trust values database or past interactions database;
2. the direct comparisons of the function of satisfaction S or the trust metric T in presence of an accepted common domain ontology;
3. the approximation of the function S or T using stereotypes situations without common ontology.

Goal

In all the strategies, the agents' goal is to get an idea of other agent's trust metric T or function of satisfaction S.

Sharing of Trust Value DB

Agents share a DB containing at least the couple $< \textit{agent name}, \ \textit{trust value} >$.
Two agents check if they have some acquaintances in common and they use common connections' trust value to compute a compatibility degree.

Pro

This strategy predicts the agents trust metric T, it does not require any knowledge of the Trust model and, according to the number of agents in common, it could became an accurate indicator.

Con

Privacy is very poor; communication overload can be heavy like the risk involved.
Solutions: adding an extra-layer of security to guarantee the confidentiality of the information shared (trusting computing & encryption keys policies).

Sharing of Evidences

Agents share a DB containing information about single interactions and how each agent evaluated each interaction.

Pro

Agents predict the function S rather than the Trust Metric T, it does not require any knowledge of the Trust model and, according to the number of agents in common, could became an accurate indicator.
Privacy concerns are less relevant since evidences are anonymous in the sense that they describe situations rather than agents' personal information.

Con

Communication overload is even bigger

Direct comparison of functions S

When there is a common ontology describing facts that is accepted by all agents, each of them can easily map its function S over this common ontology and directly compare it with the others'.

Pro

By directly comparing the two functions, agents compute an accurate degree of compatibility, without disclosing sensitive information about other agents or personal experience, and with a few communication overload (unlike the previous two solutions).

Con

It requires the strong hypothesis of a common ontology for outcome evaluations.

Predicting S and T using stereotypes situations

When there is no common ontology a solution can be build by using **stereotypes situations**. Agents' goal is to accurately predict other agent's function S using the minimum number of messages. Each agent sends stereotypes situation, considered meaningful, to the other agent and wait for its evaluation .
A strategy to generate the appropriate next situation is required and it should stop when enough information has been collected to understand other's agent model.

Pro

Sensitive data are not disclosed, communication overload is relatively small, varying from the perfect situation of solution 2 to the case where many messages have to be exchanged in order to understand other agents. (The more similar the agents' representations are, the less number of messages are required)

Con

The analysis of this issue requires further investigation beyond the scope of this paper.

Seller

Sellers are modelled as a function that defines the quality of the items he can sell (eBay-like):

- item as described $[-1,\ 1]$;
- communication $[-1,\ 1]$;
- shipping time $[-1,\ 1]$;
- shipping cost $[-1,\ 1]$;
- pricing $[-1,\ 1]$.

The seller function S is defined as an aggregation of these factors in the range $[-1, 1]$
When a buyer decides to buy from a seller, a seller will produce a n-tuple describing the quality of the item purchased. The buyer will then assess its satisfaction.

Buyer

A buyer is modelled with the following functions:

- a trust metric producing a trust value for a seller;
- a function of satisfaction that evaluates the degree of satisfaction of the buyer after purchasing an item.

Each buyer has a threshold T, the minimum seller's trust value needed to buy from it.
After deciding to buy or not, the 5 quality factors of the item sold are disclosed to understand if the buyer's decision was correct.

Each buyer has a local database of:

- sellers' trust values;
- history of its transactions;
- all the buyers known with a degree of similarity between it and other buyers.

Computation of Trust Value

The trust metric will be dependant only on direct and indirect experience.

$T_{value} : i \cdot T^{local}_{value} + (1 - i) \cdot T^{recommended}_{value}$

- T^{local}_{value}: value stored in the trust-value DB of the buyer;
- $T^{recommended}_{value}$: it is a weighted average of all of the local trust values collected from the other buyers in the community, that know that seller, using the number of interactions completed.
- i : random variable that represents in which proportion recommended and local trust values influence the final trust value.

In absence of interactions, the buyer uses its own *dispositional trust value*, i.e. a degree of ditrust. After an interaction, the local trust value is updated like this:

$T^{local}_{value} : m \cdot T_{value} + (m - i) \cdot V_{satisfaction}$

- m: value representing the effect of memory, i.e. how much the last interaction count on the new local trust value;
- $V_{satisfaction}$: explained later.

IFIPTM 2008

Value of Satisfaction

For each transaction a value of satisfaction of the buyer is computed in any case (item sold or not).
Each buyer has a vector of preferences defined as a n-tuple of value between $[-1, 1]$, uniformly distributed random variable decided at the simulation start-up, representing the importance of each factor. (1: essential: 0: not influent; -1: inverted proportional).
This vector is in correspondence with the seller's n-tuple of item quality factors.
The computation of the value of satisfaction is the scalar product between the two vectors.

Degree of Compatibility

This value is computed directly comparing the functions S of two buyers, represented by a linear combination of the 5 quality factors (eBay-like).
The computation is a scalar product among two 5-dimension normalized vectors.

How the simulation works

1. A couple buyer, seller $< B_{uyer}, S_{eller} >$ is selected randomly
2. B_{uyer} computes the trust value of S_{eller}
   - a) B_{uyer} retrieves the local trust value
   - b) B_{uyer} collects the recommended trust values of the S_{eller}, from other buyers
   - c) B_{uyer} computes the degree of compatibility for each recommending Buyer
   - d) B_{uyer} computes the aggregated recommended trust value
   - e) B_{uyer} computes the trust value for the S_{eller}
3. IF ($T_{value} >$ threshold $\geq B_{uyer}$) THEN buys from S_{eller}
   - a) S_{eller} generates item quality factors n-tuple
   - b) B_{uyer} computes its level of satisfaction
   - c) B_{uyer} updates its local trust value for S_{eller}
   - d) B_{uyer} updates its interaction DB

   ELSE: e. b. d. (trust value not updated)

When the simulation is running without the degree of compatibility, step 2c is skipped. If agents don't use recommended values, step 2b and 2d are also skipped.

IFIPTM 2008

Benefits of recommendations enhanced by a degree of compatibility

Simulation

20 sellers, 50 buyers, 1000, 5000, 10000, 20000 transactions in:

- A) buyers don't use recommendations at all;
- B) buyers use recommendations without degree of compatibility ;
- C) buyers filter recommendations using a degree of compatibility generated by directly comparing the two functions S. (a threshold of compatibility is used).

Strategy: all the agents have a function of satisfaction (combination of quality factors).
Assumption: agents are not malicious: they always transmit their real values.

Here our goal is to show that:

- when agents are different, the quality of recommendation is deteriorated and A's and B's results became very close. It may happen that A performs better than B (i.e., an agent decides better on its own);
- by using the compatibility value in C, the quality of recommendations is better than case B and "even if not always predictable" results are better than case B.

- P_{ok}. **(True Positive)** : # of transactions completed by the buyer whose level of satisfaction was more than the buyer's threshold (# of time it was a good idea to trust the seller);
- P_{no} **(True Negative)**: # of transactions correctly rejected by the buyer (# the transactions whose level of satisfaction would have been smaller than buyer's expectations);
- N_{ok} **(False Positive)**: # of transactions accepted by the buyer whose level of satisfaction was smaller than buyer's threshold (it would have been better to decline the transaction);
- N_{no} **(False Negative)**: # of transactions erroneously rejected by the buyer (the transactions whose level of satisfaction would have been greater than buyer's expectations).

The metrics represent the ability of the buyer of making good predictions in good and bad cases. We defined a summarizing metric as in the following:

$$F = \frac{P_{ok}}{N_{ok}} \sqrt{\frac{P_{no}}{N_{no}}}$$

- P_{ok} and N_{ok}: represent a real benefit or damage
- P_{no} and N_{no}: are a potential benefit or damage.

IFIPTM 2008

case A: buyers don't use recommendations at all

- *long term*: it is expected to perform very good since each agent has sufficient number of past interactions to predict correctly sellers' behaviour;
- *short term*: agents have not enough information on sellers (they follow dispositional values). Deteriorated values are expected since the first interactions have a strong blind component.

case B: buyers use recommendations without degree of compatibility

- *long term*: recommendations should perform like A;
- *short term*: it should work better than A since agents share their information.

case C: buyers filter recommendations using a degree of compatibility

- *long term*: it should tend to A's results, since agents receive recommendations only from other agents very compatible with them;
- *short term*: it should be better than A, for the effect of meaningful recommendations.

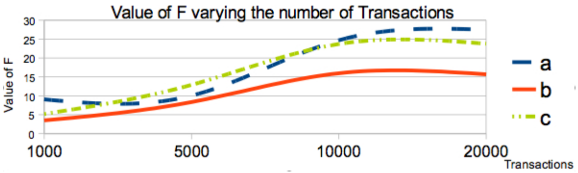
C's results should perform better than B independently from the number of transactions. Even in the long term, B should be the worst case, since the effect of recommendations, from different agents, causes deterioration in the value of recommended values.

1000 transactions

- A performs poorly as expected;
- B and C are worst than A so recommendations are not yet effective. In C, the degree of compatibility is not fully applicable yet, since it is hard to find compatible agents that interacted with the same seller.

5000 transactions

- B and C close the gap with the case A;
- C is the best case showing the effect of recommendation based on the degree of similarity;
- B is still the worst case, but the gap with the other cases is the lowest.



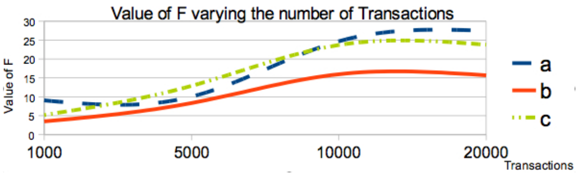
10000 transactions

- A is now the more effective, meaning that agents have gained enough direct past experience to predict sellers' trustworthiness correctly;
- C performs well, with metrics similar to A;
- B shows a growing inefficiency: recommendations without a compatibility degree are deteriorating the predictions.

20000 transactions

The three cases reach an almost steady state (similar results were obtained with 100000).

- A is the best case as predicted;
- C performs well, slightly less than A;
- B is very far from A and C.



IFIPTM 2008

Conclusions

- even in presence of strong differences among agents, the degree of compatibility, sharing common situations, keeps the quality of recommendations;
- with several recommendations, our strategy shows results comparable with a strategy purely based on past direct experience, while for an interval of interactions, our method works better than the others tested.

Future works

- investigation/evaluations of different solutions in the presence of **malicious agents**;
- a study of **trust ontology-matching** will represent an important contribution and a complementary solution to our work;
- an efficient **ontology matching** (using, for example, unsupervised techniques) can support a better computation of similarity.