AAMAS 2010
TORONTO

The 9th International Conference on
Autonomous Agents and Multiagent Systems
May 10-14, 2010
Toronto, Canada

# Workshop 29

# The Thirteenth International Workshop on Trust in Agent Societies

# TRUST-2010

Editors:
Wiebe van der Hoek
Gal A. Kaminka
Yves Lespérance
Michael Luck
Sandip Sen

# TRUST IN AGENT SOCIETIES
# (TRUST-2010)

13rd edition

Held at:
Autonomous Agents & Multi-Agent Systems Conference
AAMAS 2010
May 10th, 2010.
Toronto, Canada

Trust (along with related concepts such as privacy, reputation, security, identity) has become a major research topic in computer science. The multiagent community potentially has a lot to offer, but several conceptual and technical problems must be addressed before it can make practical contributions. Although there is increasing interest in this area within the AAMAS community, this area will need continued support as an affiliated workshop so that the AAMAS community maintains a venue for research into trust, reputation, and related topics.

## DESCRIPTION OF THE WORKSHOP

The aim of the workshop is to bring together researchers who can contribute to a better understanding of trust and reputation in agent societies. Most agent models assume trustworthy communication to exist between agents. However, this ideal situation is seldom met in reality. In the human societies, many techniques (e.g. contracts, signatures, long-term personal relationships, reputation) have been evolved over time to detect and prevent deception and fraud in communication, exchanges and relations, and hence to assure trust between agents. Artificial societies will need analogous techniques.

Trust is more than secure communication, e.g., via public key cryptography techniques. For example, the reliability of information about the status of your trade partner has little to do with secure communication. With the growing impact of electronic societies, trust and privacy become more and more important. Trust is important in applications such as human-computer interaction to model the relationship between users and their personal assistants. Different kinds of trust are needed: trust in the environment and in the infrastructure (the socio-technical system) including trust in your personal agent and in other mediating agents; trust in the potential partners; trust in the warrantors and authorities (if any). Another growing trend is the use of reputation mechanisms, and in particular the interesting link between trust and reputation. Many computational and theoretical models and approaches to reputation have been developed in the last few years. In all these cases, electronic personas may be created in many different forums (ecommerce, social networks, blogs, etc). The identity and associated trustworthiness must be ascertained for reliable interactions and transactions.

Trust appears to be foundational for the notion of "agency" and for its defining relation of acting "on behalf of". It is also critical for modeling and supporting groups and teams, organizations, co-ordination, negotiation, with the related trade-off between individual utility and collective interest;

or in modeling distributed knowledge and its circulation. In several cases the electronic medium seems to weaken the usual bonds in social control: and the disposition to cheat grows stronger. In experiments of cooperation supported by computers it has been found that people are more leaning to defeat than in face-to-face interaction, and a preliminary direct acquaintance reduces this effect. So, computer technology can even break trust relationships already held in human organizations and relations, and favor additional problems of deception and trust.

We encourage an interdisciplinary focus of the workshop - although focused on virtual environments and artificial agents - as well as presentations of a wide range of models of deception, fraud, reputation and trust building.

Just to mention some examples: AI models, BDI models, cognitive models, game theory, and organizational science theories. Suggested topics include, but are not restricted to, the following. Here "mechanisms" include considerations of architecture, design, and protocols.

- Models of trust and of its functions
- Models of deception and fraud; approaches for detection and prevention
- Models and mechanisms of reputation
- Role of control and guaranties mechanisms
- Models and mechanisms for privacy and access control
- Models and mechanisms for establishing identities in virtual worlds
- Theoretical aspects, e.g., autonomy, delegation, ownership
- Integration of conventional and agent-based mechanisms
- Policies, interoperability, protocols, ontologies, and standards
- Scalability and distribution across multiple domains or within the global domain
- Test-beds and frameworks for computational trust and reputation models
- Legal aspects
- Trust in Organizations and Institutions
- Application studies (e.g., e-commerce, e-health, e-government)

# WORKSHOP ORGANIZERS

Rino Falcone - ISTC-CNR – Italy, rino.falcone@istc.cnr.it;

Suzanne Barber - The University of Texas – USA;

Jordi Sabater-Mir - IIIA-CSIC – Spain;

Munindar Singh - North Carolina State University – USA

# PROGRAM COMMITTEE

Suzanne Barber - Computer Engineering, The University of Texas, USA

Cristiano Castelfranchi - Cognitive Science, ISTC National Research Council, Italy

Robert Demolombe - Computer Science, Institut de Recherche en Informatique de Toulouse, IRIT, France

Torsten Eymann - Department of Information Systems, University of Bayreuth

Rino Falcone - Cognitive Science, ISTC National Research Council, Italy

Wander Jager - Economics, University of Groeningen, The Netherlands

Andrew Jones - Department of Computer Science, King's College London, U.K.

Catholijn Jonker - Computer Science, Vrije Universiteit Amsterdam, The Netherlands

Churn-Jung Liau - Institute of Information Science, Academia Sinica, Taiwan

Stephane Lo Presti - Computer Science, University of Southampton, U.K.

Emiliano Lorini - Institut de Recherche en informatique de Toulouse, IRIT-CNRS, France

Steve Marsh - Computer Science, Institute for Information Technology, National Research Council of Canada

Brendan Neville - Imperial College, London, U.K.

Mario Paolucci - Cognitive Science, ISTC National Research Council, Italy

Jordi Sabater-Mir - Computer Science, IIIA-CSIC, Spain

Sandip Sen - Computer Science, University of Tulsa, USA

Munindar Singh - Computer Science - North Carolina State University, USA

Chris Snijders - Eindhoven University of Technology, The Netherlands

Eugen Staab - University of Luxembourg, Luxembourg

CONTENTS

# Modeling ART as a probabilistic planning problem with rewards

Javier Carbo and Jose Manuel Molina[1]

Group of Applied AI, Computer Science Dept., Univ. Carlos III of Madrid
{javier.carbo, josemanuel.molina}uc3m.es

**Abstract.** This paper presents a modeling of trust domain and problems, inspired in ART testbed terms and communications, using PPDDL, Probabilistic Planning Domain Definition Language. This language is an extension of the standard planning domain definition language (PDDL) that permits defining planning problems in terms of Markov decision processes in a easy-to-understand syntax. This modeling of the trust domain into a planner is open enough to define multiple very different trust models and strategies that could form a plan library to be used by BDI-like agents. We have applied it into ART testbed domain because its protocols and agent cycle provides an ordered sequence of actions, facilitating and justifying the use of planning. ART agents may decide in advance who to ask for cooperation. The possibility to anticipate these behaviours allows this approach. Future work will involve the application of this trust modeling to several cases (defined from ART competition games) in order to generate and test a collection of plans to build challenging scenarios for ART competitions.

## 1 Introduction

The way agents achieve cooperation solving complex tasks is a design key factor in MultiAgent Systems. However, since Agent Systems intend to be open, agents have to establish some kind of social control. In many real-world interactions this social control is decentralized and emergent, and it depends on local and subjective evaluations shared between partners (reputation). In recent years, trust/reputation research community has grown a lot, many trust/reputation models have been proposed [11]. Since it was difficult to compare their respective performances as many ad-hoc implementations and metrics have been applied, the Agent Reputation and Trust (ART) Testbed [6] [1] was developed. At least 20 publications used such testbed [2] and three international competitions (with 17, 18 and 11 registered teams) were successfully carried out in 2006, 2007 and 2008. So we can state that during these years the ART testbed has been used by dozens of researchers [3]. Furthermore the ART-testbed development team

[1] http://megatron.iiia.csic.es/art-testbed/

[2] http://megatron.iiia.csic.es/art-testbed/publications.htm

[3] the ART testbed discussion board yahoo group has 129 members, http://tech.groups.yahoo.com/group/art_testbed/

have discussed, patched and updated the platform using the feedback from the Competitions (see discussion notes on ART web page) and from the agent trust community (through the already mentioned discussion Board of ART). These criticism produced some changes in protocols [12], and outlined new directions of work [7]. Among them, we can remark two of them which were taken into account in this paper: the possibility of not considering participating agents as opinion providers (playing then just the role of opinion consumers), and the desirability of a set of very different scenarios where other agents were implementing predefined strategies rather than being participants in competitions.
Both directions of work were already initiated in our previous works:

- First in [1], playing games with two participants of 2007 competition and other agents defined ad hoc to be the solely opinion providers of the games, implementing honest and malicious trust behaviours. These games showed that the differences between both agents were much less than the official competition stated.
- Other of our previous publications [2] supported the second direction of work, which outlined the needing of counting on different predefined ad hoc scenarios rather than direct confrontation of competitor agents. In that paper we showed repeated games where, following an evolutive inspiration, losing agent in a game clones the winning trust strategy (replacing losing strategy by the winner one). Such repeated games has the intention to prove if the 2007 competitor winner implemented a Evolutionarily Stable Strategy (ESS in advance) and finally its strategy would become the majoritary strategy in last game. But it showed to be not a ESS, and the dominant strategy was another competitor agent (the so called 'uno', from girona univ.).

This paper follows both directions of work, since the motivation of this work is to facilitate the automated generation of different trust strategies (through the use of a plan library) to be adopted in competitive scenarios for ART games, where the correspoding agents do not act as opinion providers, just as opinion consumers.
On the other hand, the use of plans in BDI-like agents is very important, since they are assumed to respond to changes in their goals and beliefs, resulting from perception, by selecting appropriate plans and instantiating them as a collection of intentions. Although the BDI paradigm has attracted some attention from trust researchers as in [10], it has been address from the logical foundation instead of focusing in the generation and use of planning. Additionally we can remark a publication on the cooperative way to select plans in trust domain [8], but it did not address the problem of how these plans were generated. Plans are defined through a set of operators (often called actions) that, in trust domain, would address with several complex, uncertain and dynamic decisions such as who to ask for reputation and services. For instance, in [9] authors mentioned the idea of considering concatenation, aggregation and selection of reputation as operators forming a plan. All of them correspond to epistemic decisions (those about about updating and generating trust opinions from received reputations). But, as ART testbed definitions showed and cognitive theories state [3], other

additional operators have to be defined related to pragmatic and memetic decisions:

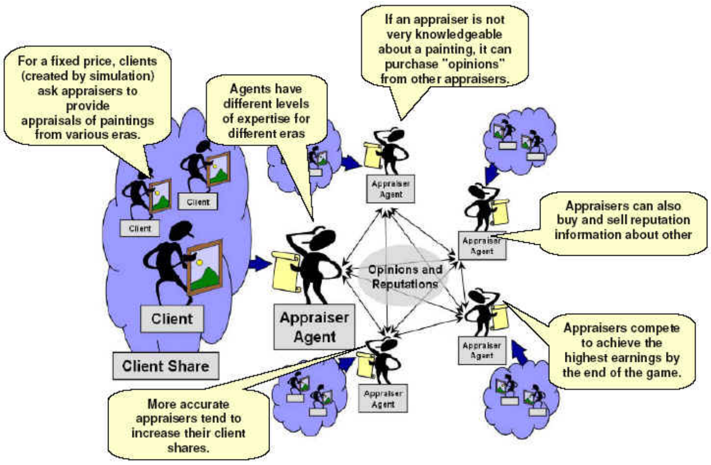
- pragmatic-strategic decisions are decisions of how to behave with partners using these reputation-based trust.
- memetic decisions stand for the decisions of how and when to share reputation with others.

This becomes an additional motivation of our work: the possibility of encapsulating of pragmatic trust decisions into operators/actions, since it would allow evaluating actions by it selfs rather than jointly (as ART testbed does). This classification has been taken into account in models such as Repage [13] and implicitly in [7] when trust model (epistemic decisions) is distinguished from trust strategy (pragmatic and memetic decisions).

## 2 ART testbed terminology and protocols

The ART testbed compared different trust/reputation models and strategies in the art appraisal domain. In this domain, agents are players/competitors that appraise paintings and implement trust strategies. Very close valuations of paintings to the real value would lead to more future clients, and therefore to more earnings to win the competition. Each painting belongs to an era among a finite set of possible artistic eras while agents have different levels of expertise (ability to appraise) in each artistic eras. An agent may request opinions from

**Fig. 1.** ART domain outline. Source [6]


If an appraiser is not very knowledgeable about a painting, it can purchase "opinions" from other appraisers.
For a fixed price, clients (created by simulation) ask appraisers to provide appraisals of paintings from various eras.
Agents have different levels of expertise for different eras
Appraisers can also buy and sell reputation information about other
Client
Client
Client
Client Share
Appraiser Agent
Appraiser Agent
Appraiser Agent
Appraiser Agent
Appraiser Agent
Opinions and Reputations
Appraisers compete to achieve the highest earnings by the end of the game.
More accurate appraisers tend to increase their client shares.

other appraisers about its paintings. With them, any agent would compute an aggregated appraisal closer to the real value (specially useful in the eras where the agent has low expertise). An agent can act also as provider of appraisals in response to opinion (about paintings) requests from other agents . Additionally, an agent can similarly request reputation information about the expertise of other appraisers on an era, and it can receive reputation requests from other appraisers on an era. All these interactions and the corresponding decisions take place in sequential order forming the iterative cycle of an appraisal agent that start after the agent receives paintings to be appraised from the simulation engine (representing assumed clients), paying a fixed fee for each appraisal request:

1. decides which eras to ask on.
2. decides who to ask for reputation on those eras
3. asks for reputation to given agents on given eras (at a reputation cost).
4. receives reputation requests from other agents
5. decides who to answer reputation requests
6. decides how to answer reputation requests
7. answer reputation requests earning the corresponding reputation cost
8. receives reputation responses
9. aggregates received reputation responses.
10. decides which paintings to ask an opinion on
11. decides who to ask for opinion on the chosen paintings
12. asks for opinion to given agents on given paintings (at a fixed opinion cost).
13. receives opinion requests from other agents
14. decides who to answer opinion requests at the corresponding opinion cost
15. decides how to answer opinion requests (investing as much as it decides)
16. answers opinion requests
17. receives opinion responses
18. appraise its own paintings (investing as much as it decides)
19. aggregates received opinion responses into a final appraisal of each painting

## 3 Expressing Planning Domains with Probabilities and Rewards

A standard domain description language, PDDL [4], is extensively used in planning community for sharing deterministic domain models and problems, enabling direct comparisons of different planning systems. A similar description language, PPDDL was proposed in [14] for the evaluation of probabilistic planning systems. A PDDL planning domain consists of a set of types, a set of objects, a set of predicates, and a set of actions:

- Objects in PDDL are terms that have some type, where there can be sub typing relationships.
- PDDL Predicates are a way to encode Boolean state variables, while functions encode numeric state variables. In order to determine the extent of the state space for planning problems, all functions and predicates take no arguments. Additionally the complete set of objects has to be known in advance.

- Actions in PDDL represent state transitions, where state stands for a particular assignment to the set of state variables of a planning problem. Actions have preconditions and effects. Preconditions characterize the states that the action is applicable in, while effects specify updates to state variables (typically increase and decrease). On the opposite way to predicates and functions, actions often have parameters.

A PDDL planning problem also includes a goal and an optimization metric that is a function of the state variables evaluated in a goal. PPDDL extends this language supporting probabilistic effects and rewards.

- Probabilistic effects declare exhaustive sets of probability-weighted outcomes, they consist of probability-effect pairs that have to sum up 1 (unless empty effect implicitly has the lacking probability). Probabilities may also appear associated with the definition of state variables of the initial state. PPDDL allows arbitrary nesting of conditional and probabilistic effects.
- Rewards in PPDDL are encoded using fluents, where the fluent reward represents the total accumulated reward since the start of the execution. Rewards are updated in action effects through the use of increase or decrease followed by a numerical expression (not involving reward). Action preconditions and effect conditions can not refer to reward (in other words, reward is not part of the state space).

Although regular PDDL goals uses an explicit optimization metric for the planning system, problems using rewards have a default plan objective that consists of maximizing the expected reward.
Further details on PPDDL syntax, semantics and examples can be found in [5]
Planning in ART testbed games depends upon several conditions (paintings belonging to eras, expertise had by own agent, expected expertise had by other agents, their certainty, ...). Since some of these conditions are expected values, they can be represented by probabilities. Additionally the goal pursued by the agent is to maximize the future reward that the agent would receive (in terms of more paintings from clients to be appraised in the next iteration). So it seems that PPDDL may be used to represent ART domain and problems.

## 4 Modeling ART games as planning domain and problems

### 4.1 Scope

First we select which steps out of the 19 ones involved in ART iteration cycle are of our interest. These 19 steps combine the behaviours of a participant agent as a provider of opinions and as a consumer of opinions. This design decision is controversial since the final earnings of participant agents come both from their ability to filter out the untrustfull partners (which is the right goal of trust models) and from their ability to exploit untrustfull behaviour. This confusion may be solved avoiding steps 13, 14, 15 and 16, and even eliminating the earnings from propagating reputation, steps 5,6 and 7 (this assumption avoids a player agent providing reputation values to other, assigning recommender role to third party -nonplaying- agents as clients should be, and opinion providers we decided to be). Step 18 has to be also eliminated since the agent is acting as a provider (to itself), and additionally we obtain agents more dependant on knowledge about the others. Therefore we still have 12 steps to be instantiated as a plan in the ART testbed domain and games. Additionally steps corresponding to send/receive messages (step 3, 8 and 12, 13) does not play any role in the plan, they could be represented as actions belonging to a plan, but their inclusion is automatic since they depend completely on decisions about asking-for-reputation and asking-for-opinion steps (1-2 and 10-11) On the other hand, steps 9 and 19 represent aggregation and concatenation operators defined in [9]. As we said before, they are epistemic actions and furthermore, they depend upon external events that can not be known in advance (the responses received from other agents). Therefore they can not be 'planned' and we did not consider them in our planning system although they should be there. In conclusion, for us, the decisions where the sequence of actions to be chosen is relevant, correspond exclusively to steps 1-2 and steps 10-11, which are the pragmactic actins we were interested in, and furthermore these steps determine the actions to be held in advance (asking for reputation and asking for opinions). Then, our goal of representing (part of) an iteration of an ART player agent as a planning problem is focused on these steps, which makes sense since almost these steps may be 'planned' from the moment we received the painting to be appraised. The existence of several paintings and providers of the same era opens the problem to very different ways to act in order to reduce the expected uncertainty of the painting final appraisal values.

### 4.2 limitations

There is one important limitation, that consists of the use of reputation into the decision of selecting the agents who ask for opinion (step 11 of section 2). Since planning requires all objects and effects to be known, the reputation responses received from other agents can not be previewed ('known') and therefore, such decision may not consider these responses in the given iteration (they will be taken into account in next iterations). Additionally, first iteration(/s) of an ART game is special since no agent has any knowledge about opinion providers and asking for reputation does not make sense at this point [4] We assume agents to be in an intermediate point of an ART game, forgetting then the specific problems of the different logic to be followed in the first iteration(/s).

[4] In fact this cold start of ART games may be biasing the competition results since the winning trust strategy in those games may be not so successful when all participating agents have some 'a priori' experience-based knowledge about opinion providers reliability.

### 4.3 definition of objects and types

The basic concepts of ART games are eras, agents and paintings [5]. We define the instantiation of these concepts as objects in the definition of the planning problem:

```
(:objects era1 era2 ... era5 - era)
(:objects iam argente2 uno2008 connected ... -agent)
(:objects painting1 painting2 ... painting9 -painting)
```

Each of them with its corresponding type era, agent and painting has to be defined in the domain file:`(:types era agent painting)`

### 4.4 definition of predicates and functions

Predicates represent the statements about the objects in the problem that are assumed to be true. One of them would represent the assignment of eras to paintings: `(requested-appraisal ?painting -painting ?era -era)`

The corresponding costs of opinions and reputation are functions opinion-cost and reputation-cost (since they encode a numeric state variable), defined in the initial state:

`(opinion-cost 10)(reputation-cost 1)`

These costs are taken into account in two ways: first, verifying that the agent can afford the cost, and therefore, we have to represent the bank balance of the agent (as a function `bank-balance`). Second, to taking into account the paid costs in the optimization goal defined implicitly by the rewards. We also need a predicate meaning how much expertise is needed on that era, computed (out of the scope of the plan) from the the number of paintings to appraise in that era:

`(needed-expertise ?era)`

Additionally we have to represent the expertise of an agent about an era, it could be a function: `(has-expertise ?appraiser -agent ?era -era)`

The certainty on that expertise is represented also by a function:

`(has-certainty ?appraiser -agent ?era -era)`

In the same way, the ability of an agent providing useful reputation values could be represented by the function: `(has-knowledge ?appraiser -agent)`

While the cooperative intention of the agent is represented by the function:

`(is-cooperative ?recommender -agent)`

These predicates and functions will appear instantiated in the initial state that represents a specific iteration of an ART game. But there will be more predicates that will be added afterwards, as the decisions would be taken in the effects of the actions. These ones would be:

- the result of decision corresponding to steps 1-2 mentioned in section 2 would be address jointly by an action. The predicate would take the form of `(to-be-asked-for-reputation ?recommender -agent ?era -era)`

[5] the own agent is not represented as an object of type agent

- the result of decision corresponding to step 10-11 mentioned in section 2 would take the form of
  `(to-be-asked-for-opinions ?appraiser -agent ?painting -painting)`

### 4.5 definition of actions

Actions in PDDL represent state transitions, with conditions and effects. Decisions of step 1-2 can be jointly addressed by an action for asking-for-reputation, where the parameters are: era, recommender and appraiser. its preconditions are the next ones:

- that our agent has enough bank-balance. In other words, it should be greater than reputation cost: `(> (bank-balance) (reputation-cost))`
  And therefore in the effects bank-balance should be decreased in reputation-cost: `(decrease (bank-balance) (reputation-cost))`
  Additionally the reputation-cost would be considered negatively in the numerical function f that computed the increase of expected reward obtained by executing this action.
- that we need expertise in that era (how much we need it). It should be greater than a minimum defined by a given threshold (represented by a function min-needed-expertise)
  `(> (min-needed-expertise) (needed-expertise ?era))`
  Additionally the needed-expertise would be considered positively in the numerical function f that computed the increase of expected reward obtained by executing this action. The more needed is the expertise in that era, the more interesting is executing this action.
- that are not very sure of the knowledge I have about a potential appraiser in an era. This is computed comparing if the certainty of the appraiser in an era is lees than a given threshold (function max-certainty). The less knowledge i have about the expertise of a potential appraiser, the more interesting is executing this action.
  `(< (max-certainty) (has-certainty ?appraiser ?era))`
  Additionally the certainty had by would be considered negatively in the numerical functions that computes the increase of expected reward obtained by executing this action (f). The more certainty has the appraiser, the less interesting is executing this action. Finally the certainty had by the potential appraiser would become increased in some level, computed as a function g of has-knowledge value, and the probability that this increment and the reward would take place depends upon the how much cooperative is the recommender (function h). This increment of has-certainty value of the appraiser would avoid an endless repetition of the same instance of this action.
- that the recommender agent is cooperative. It should be greater than a minimum defined by a given threshold (represented by a function min-cooperation)
  `(> (min-cooperation) (is-cooperative ?recommender))`
  Additionally the level of cooperation of the recommender would be considered positively in the numerical function f that computed the increase of

expected reward obtained by executing this action. The more cooperative is the recommender, the more interesting is executing this action.
- that the recommender agent is often well-informed (has knowledge about appraisers). It should be greater than a minimum defined by a given threshold (represented by a function min-knowledge)
`(> (min-knowledge) (has-knowledge ?recommender))`
Additionally the level of knowledge that the recommender probably has, would be considered positively in the numerical function f that computed the increase of expected reward obtained by executing this action. The more informed is the recommender, the more interesting is executing this action.

The effects of action of asking for reputation are:

- the corresponding decrease of bank-balance
- the inclusion of the predicate:
`(to-be-asked-for-reputation ?recommender ?era)`
which would fire the execution of the step 3, waiting then for the external event produced in step 8 of section 2.
- the increment of has-certainty value on the appraiser and the era in a function g dependant of has-knowledge value. A possible definition could be:
`(increase (has-certainty ?appraiser ?era) (has-knowledge ?recommender))`
- the increment of the reward corresponding to the execution of this action, would be computed from function f dependant on reputation-cost, needed-expertise, has-knowledge and has-certainty functions. The most simple implementation could be a sum of all of the positive ones subtracting the negative ones: `(increase (reward) (+ (+ (has-knowledge ?recommender) (needed-expertise ?era)) - (+ (has-certainty ?appraiser ?era)(reputation-cost))`

Finally, to represent the possibility of a non cooperative attitude from recommender agent, these last two effects are considered probabilistic effects, where the probability associated to them has to be dependant of is-cooperative function. An example of this probability could be: `(probabilistic (is-cooperative ?recommender) (increase (has-certainty ...)) (increase (reward) ...)))`
The other action defined in the ART planning domain is asking-for-opinion (steps 10 and 11). The parameters are: era, painting and appraiser. The preconditions of this action are the next ones:

- that our agent has enough bank-balance. In the same way that the previous action but with opinion cost rather than reputation cost.
- that our agent has to appraise a painting on the given era:
`(requested-appraisal ?painting -painting ?era -era)`
- needed-expertise compared with min-needed-expertise
`(> (min-needed-expertise) (needed-expertise ?era))`
Additionally the needed-expertise would be considered positively in the numerical function f that computed the increase of expected reward obtained by executing this action. The more needed is the expertise in that era, the more interesting is executing this action. Finally the needed-expertise in

an era would become reduced in some level, computed as a function g of has-expertise value, and the probability that this decrement and the reward would take place depends upon the how much certainty we have in the expertise of the appraiser in that era (function h). This decrement of needed-expertise value of the era would avoid an endless repetition of the same instance of this action.

- that i am very sure of the knowledge I have about a potential appraiser in an era. This is computed comparing if the certainty of the appraiser in an era is more than a given threshold (function min-certainty).
  `(> (min-certainty) (has-certainty ?appraiser ?era))`
  Additionally the certainty had by would be considered positively in the numerical functions that computes the increase of expected reward obtained by executing this action (f). The more certainty has the appraiser, the more interesting is executing this action. This is the opposite of what we considered in asking-for-reputation action. Finally the certainty had by the potential appraiser would become reduced in some level, computed as a function g, possibly defined as an uniform function.
- that the potential appraiser has enough expertise on that era. This is computed comparing the expertise of the potential appraiser with a given threshold (function min-expertise). The more expertise the agent is supposed to have, the more interesting is executing this action.
  `(> (min-expertise) (has-expertise ?appraiser ?era))`
  Additionally the expertise had by the appraiser would be considered positively in the numerical functions that computes the increase of expected reward obtained by executing this action (f). The more expertise has the appraiser, the more interesting is executing this action.

The effects of action of asking for opinion are:

- the corresponding decrease of bank-balance
- the inclusion of the predicate:
  `(to-be-asked-for-opinion ?appraiser ?era)`
  which would fire the execution of the step 12, waiting then for the external event produced in step 13 defined in section 2.
- the increment of has-certainty value on the appraiser and the era in a function g typically uniform. A possible simple definition could be: `(increase (has-certainty ?appraiser ?era) 1)`
- the increment of the reward corresponding to the execution of this action, would be computed from function f dependant on opinion-cost, needed-expertise, has-expertise and has-certainty functions. The most simple implementation could be a sum of all of the positive ones subtracting the negative ones: `(increase (reward) (+ (+ (has-expertise ?recommender) (needed-expertise ?era) (has-certainty ?appraiser ?era) - (reputation-cost)`

Finally, to represent the possibility of a deceptive result from the opinion of appraiser agent, these last two effects are considered probabilistic effects, where the probability associated to them has to be dependant of has-certainty function.

An example of this probability could be: `(probabilistic (has-certainty ?appraiser) (increase (has-certainty ...)) (increase (reward) ...)))` Obviously the definition of functions f, g, and h could be customized introducing new factors, or weighting them in diferrent ways.

### 4.6 generated plans

The execution of the defined actions in very simple art game definitions, would provide an output wich consists of a collection of predicates such as:

```
to-be-asked-for-reputation iam argente era2
to-be-asked-for-reputation iam uno2008 era3
...
to-be-asked-for-opinion argente painting1
to-be-asked-for-opinion connected painting4
...
```

Machine readable version of this domain and a couple of example problems can be found on line at
`http://www.giaa.inf.uc3m.es/miembros/jcarbo/art-planner.ppddl`

## 5 Conclusions

Modeling ART games as a planning system can not be done without many limitations. But taking into account such restrictions, part of the basic iteration of a player agent in such games can be modeled successfully as a planning domain and problem. Besides explicitly using a very descriptive language trust decisions, this planning approach is a step forward in the possibility of using BDI-like agents in this problem, and it intends to be a way to generate non trivial challenging agents (current honest and malicious ART predefined agents are mostly dumb). With them, it would be closer the goal of ART testbed development team relative to generating a complete set of artificial societies for improved competition and experimentation. Although this paper only shows how ART testbed was modeled as a planning domain and problem, future works involve evaluating the corresponding generated plans in already-played games (for instance 2006, 2007 and 2008 competitions). These evaluation would allow us to tune properly the thresholds min-knowledge, min-certainty, min-expertise and min-needed-expertise.

## 6 Acknowledgements

This work was supported in part by Projects CICYT TIN2008-06742-C02-02/TSI, CICYT TEC2008-06732-C02-02/TEC, CAM CONTEXTS S2009/TIC-1485 and DPS2008-07029-C02-02

## References

1. J. Carbo and J. M. Molina. An extension of a fuzzy reputation agent trust model (afras) in the art testbed. *Soft Computing*, DOI 10.1007/s00500-009-0470-9, 2009.
2. J. Carbo and J. M. Molina. Finding an evolutionarily stable strategy in agent reputation and trust (art) testbed. In *Proceedings of 23rd Int. Conf. on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA-AIE 2010, special session on Soft computing in information access systems on the Web*, pages Accepted, to be published in June. Lecture Notes in Artificial Intelligence, Springer-Verlag, 2010.
3. R. Conte and M. Paolucci. *Reputation in Artificial Societies.* Kluwer Academic Publishers, 2002.
4. M. Fox and D. Long. Pddl 2.1: An extension to pddl for expressing temporal planning domains. *Journal of Artificial Intelligence Research*, 20:61–124, 2003.
5. M. Fox and D. Long. The first probabilistic track of the international planning competition. *Journal of Artificial Intelligence Research*, 24:851–887, 2005.
6. K. Fullam, T. Klos, G. Muller, J. Sabater, A. Schlosser, Z. Topol, K. S. Barber, J. Rosenschein, L. Vercouter, and M. Voss. A specification of the agent reputation and trust (art) testbed: Experimentation and competition for trust in agent societies. In *The Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2005)*, pages 512–518, 2005.
7. M. Gomez, J. Sabater-Mir, J. Carbo, and G. Muller. Improving the art testbed, thoughts and reflections. In *Procs. of 12$^{th}$ CAEPIA Conference*, pages 1–15, 2007.
8. N. Griffiths and M. Luck. Cooperative plan selection through trust. In *Multi-Agent System Engineering: Proceedings of the Ninth European Workshop on Modelling Autonomous Agents in a Multi-Agent*, pages 162–174. Springer-Verlag, 1999.
9. C.-W. Hang, Y. Wang, and M. P. Singh. Operators for propagating trust and their evaluation in social networks. In *AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, pages 1025–1032, Richland, SC, 2009. International Foundation for Autonomous Agents and Multiagent Systems.
10. I. Pinyol and J. Sabater-Mir. Pragmatic-strategic reputation-based decisions in bdi agents. In *AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, pages 1001–1008, Richland, SC, 2009. International Foundation for Autonomous Agents and Multiagent Systems.
11. S. D. Ramchurn, D. Huynh, and N. R. Jennings. Trust in multi-agent systems. *Knowl. Eng. Rev.*, 19(1):1–25, 2004.
12. J. Sabater, M. Gomez, G. Muller, and J. Carbo. Changes for the 2008 competition, 2008.
13. J. Sabater-Mir, M. Paolucci, and R. Conte. Repage: Reputation and image among limited autonomous partners. *Journal of Artificial Societies and Social Simulation*, 9(2), Mar. 2006.
14. H. L. S. Younes and M. L. Littman. Ppddl1.0: An extension to pddl for expressing planning domains with probabilistic effects. Technical Report CMU-CS-04-167, School Of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, 2004.

# The Social Dynamics of Learning and Trust

Maurizio Cortesi

15th March 2010

Organization: University of Luxembourg - CREA[1]

**Abstract**

Trust is often described as an important variable in social interactions. This has been true not only for disciplines such as sociology, psychology or anthropology. Economics scholars have mainly used trust as a background characteristic, allowing exchanges and potentially influencing economic performance. In the last decades, however, also due to insights coming from other disciplines and to better available measures, economic research on trust consequences has become more and more quantitative. Simultaneously, the understanding of trust antecedents has deepened.

In this paper we present a model integrating some recent insights from experimental research, and we investigate how agents interactions based on trust may influence both individuals and population performance in terms of learning efficiency and quality of the interactions. We use a definition of trust based on economic primitives, such as beliefs and preferences, to evaluate the ability of agents to learn about the interaction environment as to choose the most fruitful relationships.

We show that preferences and learning dynamics may play an important role in shaping individuals performance.

**Keywords:** trust, preferences, beliefs, learning, social networks.

## 1 Introduction

Most of the literature on trust deals with dyadic interactions, both in theoretical and experimental approaches [2, 4, 19]. Instead, in this paper we focus on trust dynamics in a social environment.

[1] Email: maurizio.cortesi@uni.lu

This research is made possible under the financial support of the FNR fellowship project AFR PHD017

Interesting network models of trust can be found in computer science. Generally these models deal with trust in terms of reputational systems, with a particular focus on the mechanisms for the evaluation of network nodes, whether they are artificial (computers) or human [6, 3, 29, 34, 17, 21].

On another side, economists generally link social environments and trust along three main different perspectives. A first perspective goes under the broad category of social capital literature [11, 5]. Secondly, managerial literature focuses on the link between trust in groups and organizational performance [27, 12, 28]. Finally, an increasing number of studies focuses on the link between macro-economic performance and culture [23, 30, 15, 31, 18].

We believe there is a missing level of analysis in the literature, linking theoretical modeling at the dyadic level and empirical research at the social level.

More specifically, we are here interested in investigating how trust dynamics evolve in a social network. On one side trust allows for social interactions to take place; on the other side, the outcomes of these interactions allow for an update of trust itself. This is a potentially self-reinforcing mechanism, where trust is needed to generate more trust. We focus therefore our attention on the dynamics of trust learning and on individual and social performance.

The rest of the paper is structured as follows. Section 2 gives an overview on trust related research. Section 3 describes the formalization of the model. Section 4 describes performance measures and discuss the results. Finally section 5 highlights some possible conclusions.

## 2 Trust literature

Trust is central to all transactions and yet economists rarely discuss the notion. It is treated rather as background environment, present whenever called upon, a sort of ever-ready lubricant that permits voluntary participation in production and exchange [11].

Over than this, some authors correctly claim, that most of the research on this topic simply repeats the same results or implications of studies on transaction costs or on Prisoners dilemma-like situations [9]. Instead, a cognitive theory of trust is needed as part of an enriched description of the complex structure of beliefs and goals that are at the basis of agents choices and actions, being these choices and actions socially embedded and strongly influencing agents also in the economic arena.

In particular, an interesting social aspect of trust is that related to its risk component. In fact, trust appears as a solution to specific problems of risk, that is the risk of being disappointed by others actions and suffering a damage bigger than the advantage one was seeking choosing to trust [26].

Some of these aspects are better understood when one looks to recent studies, linking trust and its mental components in terms of beliefs and preferences. Fehr [13] documents the accumulation of strong evidence that trusting cannot be captured by beliefs about other people's trustworthiness and risk preferences alone, but that social preferences play a key role.

In example, betrayal aversion, that suggests that people are more willing to take risk when facing a given probability of bad luck than to trust when facing an identical probability of being cheated, seems to play a particularly important role in trusting behavior [7]. Kosfeld et al. [24] study the effects of oxytocin, an hormone that has been described as an important enhancer of pro-social and affiliation behavior in many different non-human mammals. It is shown that this substance may also increase human pro-social behavior and evidence is given that trust relates to risk generated in a social environment and in social interactions. Also studies on biology and evolution are also showing that this behavior might be codified and appear in specific ways.

Trust, for instance, is a possible solution to problems arising from uncertainty, risk, incomplete information, incomplete knowledge, imperfect rationality. As trust diffuses and is practiced, agents get better and better in understanding and predicting the environment in which they act.

As Good [16] suggests, the reaction of another agent to one's own action is important in confirming prior experience, and this confirmation (predictability) is necessary to make the social world intelligible and seemingly knowable. Though some prerequisites are necessary for these dynamics to be at work. Firstly agents need to interact in an environment rich of trust and trustworthiness, since trust may be depleted through not being used [14]. Secondly, agents need some capabilities and rules to discriminate among their potential counterparts and a sufficiently developed information processing ability and flexibility [25] as to make it possible to choose the best partners to place trust in.

Signaling and the ability to decode correctly and profitably these signals, are therefore two important aspects of social interactions in general, and trust based ones in particular. Signals may come not just from direct perception and experience of the environment, but also from third-party experiences diffused along the network. These informations may generate a public record of every agent that is available to all network nodes. This individual record is generally defined as an agent's reputation.

These aspects have also a strong economic appeal, and they are one of the focus topics in the research area of markets, products quality and firm reliability [1, 32].

However, very interesting insights and results come from computer science, specifically from referral systems studies or trust management studies. The problem at hand is the possibility to assess the trustworthiness of every node in the network, by means of scoring from other nodes in the same network, as to create a reputation building mechanism.

Researchers discuss important questions related to scores generation, discovery and aggregation [34]. Some underline that a trust model needs to take into account many sources of information, as to be robust against some possibly missing sources or lying from other agents. At the same time, every agent should be able to evaluate and pool all these informations on his own [21]. In these contexts, misbehavior is not only intended in the sense of not respecting the rules of the interaction, but, more interestingly, not respecting the rules of a fair scoring mechanism [33].

The presented model in the next section develops on these building block, with some important additions. Firstly, it is a model of trust in a social environment, considering more than dyadic and triadic interactions. Secondly, it uses insights from different disciplines and approaches, both for agents characterization and for the definition of interaction rules. Lastly, its simple implementation might allow extensions for more specific and deeper evaluations.

# 3 Model

We consider a network of $N$ agents, indexed as $i = 1, 2, \ldots, N$. Time is discrete, indexed with $t = 0, 1, \ldots, T$. For each agent we define a number of characteristics, over which we model interaction and learning dynamics.

Own information is embedded in a $\beta$-distributed belief $p_{ij}^t \sim \beta\left(u_{ij}^t, v_{ij}^t\right)$, that represents the information about the outcomes of past interactions between agents $i$ and $j$. In our definition of the beta function, $u_{ij}^t$ and $v_{ij}^t$ define respectively successes and unsuccesses. Based on this function, the expected likelihood of success in a partnership between $i$ and $j$ is $p_{ij}^t = u_{ij}^t / \left(u_{ij}^t + v_{ij}^t\right)$.

Every period partnerships are formed based on agents beliefs, just defined, and preferences. We define the social component of individual preferences as a threshold value $\theta_i \in [0, 1]$. This determines the level of belief over which each agent is willing to propose or accept an interaction.

As we consider exogenous preferences, these thresholds are given and do not change over the considered period. However, it may well be the case that preferences evolve over time, in response to new information and new experience [16, 22]. Moreover, this formalization resumes two insights from recent research. Firstly, these thresholds represent an indicator of the pro-social attitude of each agent [10, 24]. Secondly, they represent an indicator of agents aversion of socially-embedded risk, such as betrayal aversion [7].

Once partners are selected and interactions are made, a network $g^t$ of all the links $ij$ for which $E\left[p_{ij}\right] \geq E\left[\theta_i\right]$ and $E\left[p_{ji}\right] \geq E\left[\theta_j\right]$, where the former in each inequality defines the expected payoff given agents beliefs, and the latter the threshold payoff given agents preferences.

The interaction fails with probability depending on the level of trustworthiness of the counterparts. Trustworthiness $w_i$ is also defined in the interval [0, 1] and is stable over time. We recognize that, as for preferences, this is a strong simplified assumption. In fact, trustworthiness may evolve due to the environmental conditions and the partnering and success history of each agent. Moreover, an agent's beliefs and preferences might also have a relation with his own trustworthiness, and placing trust may influence counterparts trustworthiness in ongoing interactions. We will investigate these aspects in future developments of the model.

Back to the model, the subset $o^t \subseteq g^t$ defines the indicator of positive outcomes for each agent. An interaction is successful if and only if the outcome is positive for both the interacting agents. Otherwise, if for $i,\ j \in o^t$, $o_{ij}^t \neq o_{ji}^t$, then the interaction fails. This may allow for potentially contrasting information to flow in the network and, eventually, cheating in evaluation and communication [33].

From one period to the next, the belief $p_{ij}^t$ is updated according to Bayes rule, using the information embedded in $o^t$. After period t has taken place, positive and negative outcomes are recorded and used to update individual beliefs about partners. Therefore, $p_{ij}^{t+1} \sim \beta\left(u_{ij}^t + o_{ij}^t,\ v_{ij}^t + 1 - o_{ij}^t\right)$.

The condition for a link to be created at each period $t$ is thus

$$\pi_i^{g^t} - \pi_i^{g^t - ij} = E\left[p_{ij}^t \mid g^s,\ o^s;\ s < t\right] \geqslant E\left[\theta_i\right]$$

Given the update mechanism described, this condition is only depending on $i$ and $j$'s interactions. Although we want the model to include other $k$'s experiences. This is where third party information pooling comes in.

To generate learning from others experiences we use insights from Bayesian model averaging [20]. This is a way to evaluate the probability of a model being true, given the realizations and assessing the probability of other models too being true, given the same realizations. In our use, models are the different beliefs agents hold about someone's trustworthiness.

In the general BMA framework, if $\Delta$ is the quantity of interest, such an effect size, a future observable or the utility of a course of action, then its posteriors distribution given data D is an average of the posterior distributions of each considered model weighted by their posterior model probability. All the probabilities are conditional on the set of all models being considered. In our approach, in particular, a "model" $M_k$ is agent $k$'s belief about another agent (say $j$), that is $p^t_{kj}$. When considering also third-party information, thus, the posterior belief at time $t$ is:

$$E\left[p^t_{ij} \mid g^s,\, o^s;\, s<t\right] = E\left[p^t_{ij} \mid g^s_{ij},\, o^s_{ij};\, s<t\right] Pr\left\{M_i \mid g^s_{ij},\, o^s_{ij};\, s<t\right\} +$$

$$+ \sum_{k\in g_i, g_j} E\left[p^t_{kj} \mid g^s_{kj},\, o^s_{kj};\, s<t\right] Pr\left\{M_k \mid g^s_{kj},\, o^s_{kj};\, s<t\right\}$$

where the first terms are as defined above, while the second terms are as defined in Bayesian model averaging.

There are two important assumptions regarding the pooling of informations. Firstly, own information is weighted against third-party information, through a parameter $\alpha_i \in [0,\ 1]$, that is an individual heterogeneous character. Secondly, third-party information is assumed to be equally weighted over all the sources. Research on reputation systems underlines that possibly third-party information might be aggregated in a way to reflect the confidence level in the different sources [8]. Therefore a general form for a source related weight might be

$$w^t_k = \frac{p^t_{ik}}{\sum^m_{k\in g_i, g_j} p^t_{ik}}$$

that is, information is weighted proportionally to the belief in each source.

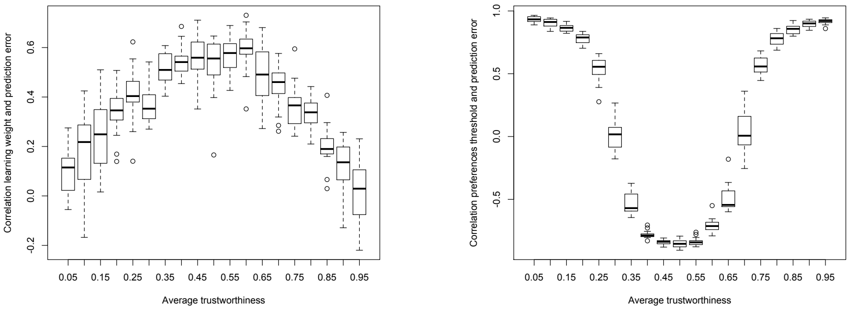
In this paper we anyhow assume all sources to be equally trustworthy. The motivation to this assumption resides in the way we model information diffusion. We allow only information about individual outcomes to flow in the network and we assume that there is no strategic use of information by agents and therefore no possible cheating in communication.

# 4 Numerical experiments and results

We run numerical experiments to investigate the effects of learning and preferences on individual and social performance at different levels of population average trustworthiness. Specifically, we consider a population of 80 agents that may interact over a time span of 60 periods. Learning weights, social preferences, beliefs are randomly initialized and are therefore heterogeneous among agents.

On one side we investigate learning dynamics. In fact, the faster the learning process, the better the chances to interact only with good partners and the more the information flowing in the network is relevant for the detection of untrustworthy counterparts. On the other side, since social preferences directly influence the number of ongoing interactions, we evaluate how this interacts with learning dynamics and the ability to choose more successful partners.

The learning performance indicator we use is the average trust prediction error, defined as the average of each agent's beliefs distance to his counterparts real trustworthiness levels after each period. We evaluate the correlation with both individual learning weights and social preferences.



(a) Learning weights

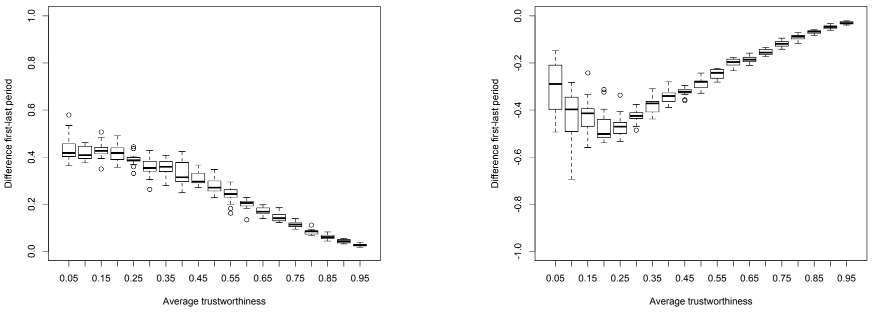
(b) Preferences thresholds

Figure 1: Prediction error

We found particularly clear patterns in these correlations. In particular, figure 1a shows that the closer the average trustworthiness of the population is to intermediate values ($\simeq 0.5$), the more relying on indirect experience is important for a better prediction of counterparts trustworthiness. Figure 1b shows instead that for lower and higher average population trustworthiness, higher preferences imply bigger prediction errors. On the contrary for intermediate values of population trustworthiness higher preferences tend to imply smaller prediction errors.

These effects appear contrasting. On one side it seems that, since for intermediate population trustworthiness indirect learning is more important, agents would need more interactions; on the other side, the negative correlation with preferences implies that the better predicting agents are those with less interactions. However, since learning is more ambiguous for intermediate levels of average population trustworthiness, counting on less interactions may be positive, as well as sharing the information of fewer interactions with a smaller number of partners. In conclusion, we believe these effects combine to reduce ambiguity.

We also evaluate the evolution of interactions along the considered time span, as well as the dynamics of positive versus negative outcomes. If we observe the dynamics of interaction evolution in time (Figure 2a) we clearly see that the percentage of interactions is always decreasing, though decreasingly for increasing population trustworthiness.



(a) Interactions

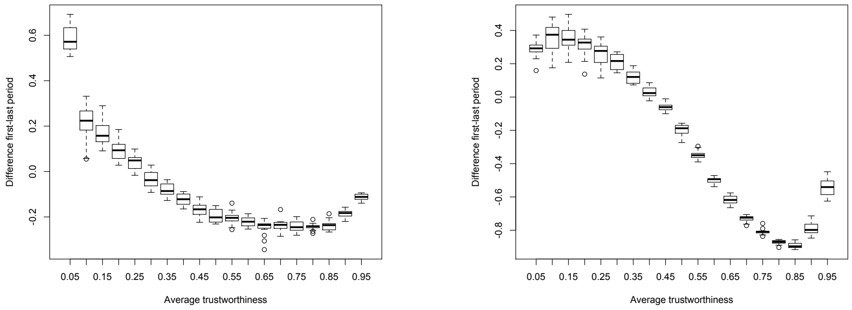
(b) Positive outcomes

Figure 2: Evolution over time - Difference first and last period

However, at the same time the percentage of positive outcomes increases (Figure 2b), showing therefore that agents are able to choose and maintain over time generally more trustworthy counterparts, either this is due to preferences thresholds or learning.

Moreover, we define a measure of the social risk and a measure of the loss due to individual thresholds. The latter is defined as the percentage of potentially fruitful interactions, those for which a counterpart's trustworthiness is higher than an agent's belief, that are not exploited due to agents thresholds, over the total of the available fruitful interactions. The former, somehow symmetrically, is the percentage of accepted and ongoing interactions for which the counterpart's trustworthiness is lower than an agent's belief, with respect to the total of ongoing interactions. In this first case, preferences are too high to allow fruitful interactions, while in the second case preferences are too low to protect the agent from misbehaving agents.

Figure 3a shows that over time the percentage of interactions lost due to individual preferences thresholds increases, though the size of this effect is small and slightly higher for higher population average trustworthiness. However, even this small effect means that agents tend to loose potentially fruitful interactions. For lower trustworthiness values we observe a decrease, but in this situation the effect is positive due to the fact that preferences protect agents from anyhow generally riskier interactions (being average trustworthiness really low) and agents apparently learn the correct trustworthiness values of their counterparts.



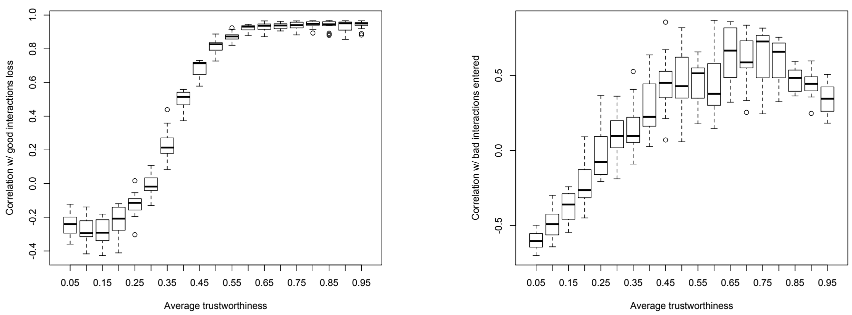
(a) Good interactions loss

(b) Bad interactions entered

Figure 3: Evolution over time - Difference first and last period

On the other side, the percentage of interactions with untrustworthy counterparts over the ongoing interactions tend to diminish, at least for lower values of population trustworthiness (Figure 3b). However, for higher values of trustworthiness the relation is inverted. Since in this situation the number of interactions doesn't change that much over time, this means that agents tend to overestimate their counterparts trustworthiness, and this, in turn, implies an increased chance of entering potentially riskier interactions with agents whom levels of trustworthiness have not been correctly assessed.

If we analyze the role of preferences on these two last variables in terms of correlations, we observe that preferences appear to be increasingly correlated to the number of lost chances for increasing population trustworthiness (Figure 4a). The negative correlation observed for lower trustworthiness values is not really indicative due to the fact that, at those values, the number of fruitful chances is anyhow really low.



(a) Good interactions loss

(b) Bad interactions entered

Figure 4: Preferences role - Correlations

Lastly, figure 4b shows the role of preferences with respect to the social risk attitude of agents. For lower population average trustworthiness preferences protect agents from entering riskier interactions. For higher values of population trustworthiness, on the contrary, preferences seem positively correlated with the level of social risk agents are taking. Even though agents with high preferences tend to be less prone to accept interactions, what seem to happen here is that these same agents come to a point where they might interact only with counterparts they more or less strongly overestimate. Apparently agents with lower preferences are better in diversifying their chances entering more interactions, and in such way they lower their social risk.

## 5 Conclusions

To understand trust one needs not only to assess its consequences. Research also investigates the link of trust with its antecedents, such as preferences and beliefs. It is exactly this that, in turn, may motivate the impact of trust on agents decision processes and on their social and economic outcomes.

In this paper we built a social model of trust along these lines. Agents trust has been defined in terms of beliefs and preferences, allowing for the implementation of recent experimental results.

Our numerical experiments give some interesting insights on trust dynamics. In particular, we showed that preferences may strongly affect individuals performance. Higher social preferences thresholds imply a lower attitude towards interactions and this, in turn, may generate lock-in effects that prevent both efficient learning and potential gains. Our results show in fact that higher preferences imply in general higher prediction errors, for instance not allowing agents to correctly evaluate their environment and exploit potentially fruitful interactions. Moreover, we also show that, not intuitively, higher preferences may even imply an higher social risk in highly trustworthy environments.

We recognize some simplifications of the present model and the need for future extensions. In particular, we intend to investigate the effects of time-varying preferences and of possible cheating in experiences communication among agents. Also an analysis of the co-evolution of trust and trustworthiness seems to be needed, due to the various ways these variables may interact and influence each other.

## References

[1] G. A. Akerlof. The market for" lemons": quality uncertainty and the market mechanism. *The quarterly journal of economics*, pages 488–500, 1970.

[2] R. Axelrod and W. D. Hamilton. The evolution of cooperation. *Science*, 211(4489): 1390–1396, 1981.

[3] S. Ba and P. A Pavlou. Evidence of the effect of trust building technology in electronic markets: Price premiums and buyer behavior. *MIS quarterly*, 26(3): 243–268, 2002.

[4] J. Berg, J. Dickhaut, and K. McCabe. Trust, reciprocity, and social history. *Games and Economic Behavior*, 10(1):122–142, 1995.

[5] N. Berggren and H. Jordahl. Free to trust: Economic freedom and social capital. *Kyklos*, 59(2):141–169, 2006.

[6] R. Bhattacharya, T. M. Devinney, and M. M. Pillutla. A formal model of trust based on outcomes. *Academy of Management Review*, 23:459–472, 1998.

[7] I. Bohnet, F. Greig, B. Herrmann, and R. Zeckhauser. Betrayal aversion: Evidence from brazil, china, oman, switzerland, turkey, and the united states. *American Economic Review*, 98(1):294–310, 2008.

[8] S. Buchegger and J. Y. Le Boudec. The effect of rumor spreading in reputation systems for mobile ad-hoc networks. volume 4, page 4.1, 2003.

[9] C. Castelfranchi and R. Falcone. Trust is much more than subjective probability: Mental components and sources of trust. volume 6, pages 10–12, 2000.

[10] J. C. Cox. How to identify trust and reciprocity. *Games and Economic Behavior*, 46(2):260–281, 2004.

[11] P. Dasgupta. Trust as a commodity. In D. Gambetta, editor, *Trust: Making and Breaking Cooperative Relations, electronic edition, Department of Sociology, University of Oxford*, pages 49–72. 2000.

[12] K. T Dirks and D. L Ferrin. The role of trust in organizational settings. *Organization Science*, 12(4):450–467, 2001.

[13] E. Fehr. On the economics and biology of trust. *Journal of the European Economic Association*, 7(2-3):235–266, 2009.

[14] D. Gambetta. Can we trust trust. In D. Gambetta, editor, *Trust: Making and Breaking Cooperative Relations, electronic edition, Department of Sociology, University of Oxford*, pages 213–237. 2000.

[15] E. L Glaeser, D. I Laibson, J. A Scheinkman, and C. L Soutter. Measuring trust. *Quarterly Journal of Economics*, 115(3):811–846, 2000.

[16] D. Good. Individuals, interpersonal relations, and trust. In D. Gambetta, editor, *Trust: Making and Breaking Cooperative Relations, electronic edition, Department of Sociology, University of Oxford*, pages 31–48. 2000.

[17] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. pages 403–412, 2004.

[18] L. Guiso, P. Sapienza, and L. Zingales. Trusting the Stock-Market. Technical report, January. Mimeo, 2007.

[19] R. Hardin. Gaming trust. In E. Ostrom and J. Walker, editors, *Trust and Reciprocity: Interdisciplinary Lessons from Experimental Research*, pages 80–101. Russell Sage Foundation, 2003.

[20] J. A. Hoeting, D. Madigan, A. E. Raftery, and C. T. Volinsky. Bayesian model averaging: a tutorial. *Statistical Science*, 14(4):382–417, 1999.

[21] 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.

[22] B. King-Casas, D. Tomlin, C. Anen, C. F Camerer, S. R Quartz, and P. R Montague. Getting to know you: Reputation and trust in a Two-Person economic exchange. *Science*, 308(5718):78–83, 2005.

[23] S. Knack and P. Keefer. Does social capital have an economic payoff? a Cross-Country investigation*. *Quarterly Journal of Economics*, 112(4):1251–1288, 1997.

[24] M. Kosfeld, M. Heinrichs, P. J Zak, U. Fischbacher, and E. Fehr. Oxytocin increases trust in humans. *Nature*, 435(2):673–676, 2005.

[25] R. Kurzban. Biological foundations of reciprocity. In E. Ostrom and J. Walker, editors, *Trust and Reciprocity: Interdisciplinary Lessons from Experimental Research.* Russell Sage Foundation, 2003.

[26] N. Luhmann. Familiarity, confidence, trust: Problems and alternatives. *Trust: Making and Breaking Cooperative Relations, electronic edition, Department of Sociology, University of Oxford*, pages 94–107, 2000.

[27] R. C. Mayer, J. H. Davis, and F. D. Schoorman. An integrative model of organizational trust. *Academy of management review*, pages 709–734, 1995.

[28] B. McEvily, V. Perrone, and A. Zaheer. Trust as an organizing principle. *Organization Science*, pages 91–103, 2003.

[29] L. Mui, M. Mohtashemi, and A. Halberstadt. A computational model of trust and reputation. pages 188–196, 2002.

[30] R. La Porta, F. Lopez-de-Silanes, A. Shleifer, and R. W Vishny. Trust in large organizations. *The American Economic Review*, 87(2):333–338, 1997.

[31] G. Tabellini. Culture and institutions: Economic development in the regions of europe. *Working paper*, 2005.

[32] J. Tirole. A theory of collective reputations (with applications to the persistence of corruption and to firm quality). *Reviw of Economic Studies*, 63:1–22, 1996.

[33] A. Whitby, A. Josang, and J. Indulska. Filtering out unfair ratings in bayesian reputation systems. *The Icfain Journal of Management Research*, 4(2):48–64, 2005.

[34] B. Yu, M. P. Singh, and K. Sycara. Developing trust in large-scale peer-to-peer systems. pages 1–10, 2004.

# Building and Using Social Structures in Agent ART

Elisabetta Erriquez, Wiebe van der Hoek, and Michael Wooldridge
{erriquez, Wiebe.Van-Der-Hoek, mjw}@liverpool.ac.uk

Agent ART,
Department of Computer Science, University of Liverpool, United Kingdom

**Abstract.** This work is based on our conjecture that agents who make decisions in scenarios where trust is important, can benefit from the representation of a *social structure*, representing the social relationships that exist between agents. We propose a concrete way of how agents can gradually *build and update* such a structure while interacting in the environment of the Agent ART Testbed. We also implement and validate the use of such a structure in the testbed: we have taken an existing competitor (Simplet) and compare its behaviour with a copy of that competitor who uses the social structure (SocialSimplet). In our documented experiments, SocialSimplet does better as regards to the quality of the interactions, the number of games won, and the total utility gained.

**Key words:** Trust, multi-agent systems, social structure

## 1 Introduction

Trust and Reputation are a key issue in the area of multi-agent systems. Like in human societies, agents have their own objectives; therefore the most rational strategy for an agent is to maximize its own utility. Like in the prisoner's dilemma [1], the rational strategy might be to defect. This leads to the important issues of trust and reputation.

An important question is what sources agents use to build their trust of others upon. Agent $a$ can base his trust or reputation of agent $b$ using experience of previous interactions between the two, or agent $a$ can ask a witness $c$ about his opinion regarding $b$. An important third source of trust, we agree with [2], is to use information about the social relationship (here called the *social structure*) between agents. If $a$ and $b$ are competing for the same resources, for instance, this may negatively affect the way they trust each other. Similarly, if agents $a$ and $b$ are likely to have complementary resources, and their cooperation would benefit both, is seems likely that they would be more inclined to trust each other.

Sabater and Sierra extend in [2] the **Regret** system with a notion of social structure in its reputation model. They then sketch a scenario demonstrating how such an enriched system would work. However, as far as we know, *implementing* social structures, and hence properly *evaluating* their added value and *validating* them, has not been done. And, most importantly, [2] does leave the issue of how a social structure *evolves*, aside. And these are exactly the issues we address in this paper.

In order to compare different implementable models (and their possible extensions) for trust in agent communities, and to provide an experimental standard, the ART testbed [3] was proposed. This testbed models an art appraisal context, where agents can offer their expertise to rate a painting, or give their opinion about the expertise and trustworthiness of other agents. ART is now a well established platform for researchers in trust to benchmark their specific trust models. We took one of the agents, Simplet ([4]) that participated in the ART competition, and cloned it to SocialSimplet enhancing it with a way to build and maintain a representation of a social structure. Then, we ran a number of competitions in which, among others, both Simplet and SocialSimplet participated, and we evaluated and compared their performance. In particular, we measured the number of successful interactions for both of them, the gained utility and we counted the number of games won by them. On all of those measures, SocialSimplet did beat Simplet, which encourages us to explore our social structures further, and to extend our experiments to other agents and other scenarios.

In summary, in this paper we (i) describe how agents can gradually build and update a representation of the social relationships and structure of their society, (ii) provide an implementation of such concept of social structure and (iii) test and analysing the result of the use of such a structure in a trust model. The remaining parts of this paper are organised as follows. Section 2 gives a further introduction and definition of social structure and describes the scenario used for the evaluation. Section 3 explains the process used by the agent to build and update the representation of the social structure. Section 4 shows the evaluation and provides a discussion about the results. Finally Section 5 presents some conclusions and future work.

## 2 Social Structure and Running Example

Autonomous agents use trust and reputation to minimise the uncertainty associated with agent interactions. In [5], it is argued that a robust trust model should integrate a number of information sources to produce a comprehensive assessment of an agent's likely behaviour. Usually agents gather and compute trust information from the direct interactions they have with each other. Although direct interactions are the most reliable source of information, information about them may not always be available. Therefore, the agent might not be able to form an opinion, based just on direct experiences, on every agent in the society without running the risk of incurring losses.

Information gathered from direct interaction can be complemented with reputation information from third party agents, called witnesses. However, the agent cannot be sure that the witnesses providing the information are doing so truthfully. This could be the case because the other agents are lying or because the information they have is inaccurate. Also agents could hide information, because they might gain an advantage in doing so. In these cases, social information about the community could be very useful. A number of trust and reputation models have been proposed for multi-agent systems [2, 6, 5] in an attempt to integrate the social dimension in the computation of a more complete trust value. However, these models suffer from some limitations. For example, it is not explained how agents can build the structure representing the social dimension, it is assumed they already possess this information. Moreover, the concept of using information inferred from the social dimension to enhance a trust model has never been implemented and validated. Social network analysis is the study of social relationships between individuals in a society, defined as a set of methods specifically developed to allow research and analysis of the relational sides of these structures [7]. These social networks are represented using graphs called *sociograms*. A different sociogram is normally built for each type of social relationship examined, for example *kinship* or *friendship*. According to the relationship examined the graph can be directed or non-directed, with or without weighted edges. However, for the purpose of this paper, we will use a simplification of these multiple sociograms. We will use a single sociogram that we call *social structure*. In the next section, we will explain how we take into account the different types of social relationships in the social structure.

In the multi-agent systems context, social structure analysis can assume a crucial role. The search for relevant information involves finding the right sources, for instance, the agents who have the required information or expertise. Thus, social network analysis is an important tool in discovering these relevant information sources. The interactions between individuals in the society suggest the addition of a new link in the social structure. Agents are aware of the presence of other agents in the society because of their direct interactions, using services or asking opinions. However, interactions and recommendations are also useful in predicting the relationships among agents. An agent can infer from an indirect recommendation that the witness agent has used the target agent, the agent the recommendation is about, as a service provider. This is because it is assumed that if the witness agent has an opinion about the target agent, it is because they have interacted in the past. Although the aim of the recommendation request is to evaluate the trustworthiness of the target, the agent is also able to draw a link, in its social structure, between the witness agent and the target agent, thus building a more complete view of its environment.

The scenario for this running example is the Agent Reputation and Trust (Art) Testbed [3]. The ART Testbed game enables to compares different strategies of agents as they act in combination. We decided to use the Art Testbed as an experimental platform because it covers relevant trust research problems and unites researchers towards solutions via unified experimentation methods, so it is a versatile experimentation tool. Moreover it provides objective metrics through which it is possible to compare and validate different trust models.

The context of the testbed is the art appraisal domain in which agents function as painting appraisers with varying levels of expertise in different artistic eras. Each appraiser agent works in competition against every other agent in the system. Clients request appraisals for paintings from different eras; if an appraiser agent does not have the expertise to complete the appraisal, it can request, at a price, opinions from other appraiser agents. Appraiser agents may also purchase from each other reputation information about other appraisers. Appraiser agents must decide when, and from whom, to request opinions and reputation information to generate accurate appraisals for clients. Appraisers receive more clients, and thus more profit, for producing more accurate appraisals. The winning appraiser agent is selected as the appraiser with the highest bank account balance.

## 3 Building the Social Structure

The concept of using a social structure to enhance a trust model has been initially proposed in Regret [2]. In Regret the agent owns a set of sociograms that show the social relations in the society. These sociograms are not necessarily complete and accurate. However, Regret does not propose a way for the agent to build the sociograms.

The main contribution of this paper is to show how the social structure can be built by each agent in the society and show, via evaluation, how using the social structure in a trust model can improve the performance of the trust model. Using the Agent Art Testbed, we are able to use one of the agents participating in the past competition and enhance its internal trust model with the social structure. As the agent interacts with other agents it gathers information about interactions and relationships to build the network of agents and to better understand its social environment.

As in Regret [2], we consider three main types of social relationship

- Competition (comp). This is the type of relation found between two agents that pursue the same goals and need the same (usually scarce) resources. This is the kind of relation that could appear between two sellers that sell the same product or between two buyers that need the same product.
- Cooperation (coop). This relation implies exchange of sincere information between the agents and some kind of predisposition to help each other if possible. In other words, we assume that two agents cannot have at the same time a competitive and a cooperative relation.
- Trade (trd). This type of relation reflects the existence of commercial transactions between two agents and is compatible either with cooperative or competitive relations.

To identify the type of relationship between two agents, we use the concept of expertise. The Art Agent Testbed assigns to agents different levels of expertise for each art era. The agents, during the game, can ask each other information about their levels of expertise. The agents are providers of a service which is the evaluation of a painting. However, they are, at the same time, consumers of this service, when their level of expertise is not good enough to evaluate a particular painting. This means that agents with similar expertise compete in the society for the same market share. Agents with different expertise are more likely to cooperate because they will need each other's services.

Therefore, the social structure is defined as a undirected weighted graph $G = (V, E)$ where

- $V$ = finite set of vertices that represent the agents in the society,
- $E \subseteq V \times V$ , is the set of the edges that correspond to links between agents, and
- each $(x, y) \in E$ has associated with it a weight $w_{xy}$

To define the weight associated with each edge we introduce the concept of *Expertise Distance*. To compute how similar or different the levels of expertise between two agents are, we use the Euclidean distance. Each era is considered as a dimension in an $N$-dimensional Euclidean Space. The number of different eras are defined in the Art Agent Testbed games setting. The values of the different levels of expertise mark points in the Euclidean Space. We can, therefore, define an agent $x$ as a vector $v = x_1, x_2, \ldots, x_n$ where $x_i \in [0, 1]$ is the level of expertise for era $i$, $i \in [1, n]$ and $n$, which is the total number of different eras, is set in the game configuration settings.

The Euclidean distance ($d$) between two $n$-dimensional vectors, or agents, is given by:

$$d = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2}$$

This distance is called in our model *Expertise distance* and it is the weight assigned to the edge linking the agents $x, y$.

In the case where the agent does not have information about the expertise of the other agent in a particular era, it is assumed the agents have the same expertise. If the agent assumes that the unknown value is the low end of the range of expertise's value, this might artificially increase the distance measure. Setting it to the same value will result in the annulment of that dimension in the distance measure.

Agents who are a small distance apart have similar expertise and are considered to be in a social relations of *competition*. The further apart the agents are in the space, the more they depend and cooperate with each other, finding themselves in a social *cooperation* relationship. The *trade* relationship is considered to be when the distance value between the agents in the Euclidean Space is in between a certain range of values. These values will be a parameter in the evaluation.

The steps used by the agent to build the social structure are the following. Every time the agent directly interacts with another agent, asking an opinion about a painting appraisal, an edge is created between the two agents. Every time the agent receives reputation information from a witness agent, about a target agent, an edge is created between the witness agent and the target agent. This is because, as mentioned before, it is assumed that an agent can infer from an indirect recommendation, or reputation information, that the witness agent has used the target as a service provider. The edges linking the agent that owns the social structure and other agents are a result of direct interactions. The edges linking the other agents in the social structure are the result of indirect interaction, or recommendation.

The use of the social structure in the trust model is inspired by Regret [2]. However the aim of the experiments is to analyse the impact of the social structure on a general trust model. Therefore the trust model of the agent selected is not modified according to the Regret trust model. Its trust model is integrated with the use of the social structure as follows. The social structure is used to find witnesses to ask for reputation information about other agents, to decide which witnesses will be consulted, and how to weight those witnesses' opinions. The first step to calculate a witness reputation is to identify the set of witnesses (W). The initial set of potential witnesses is the set of all agents that have interacted with the target agent in the past. This set, however, in a large society, can be very big and agents with frequent interactions are likely to have a considerable amount of shared information that tends to unify their way of thinking [8]. Afterwards, this initial set is filtered according to the following steps:

- To identify the components of the graph. A component is defined as a maximally connected subgraph.
- To find the set of cut-points (CP) for each component. A cut-point is a node whose removal would increase the number of components between whom no communication can travel. Therefore a cut-point can be seen from a sociological point of view as indicating some kind of local centrality. In our context it can be considered as the agent centralizing the biggest quantity of information among the agents in the components it connects.
- For each component that does not have cut-points, to choose as a representative for that component we take the node with the larger degree. If there is more than one node with the maximum degree, choose one randomly. This point is called a central point. The degree can be regarded also as a measure of local centrality; this set of nodes is named LCP (Local Centrality Points).
- The set of selected nodes is the union between the set of cut-points and the set of LCP. That is, $W = CP \cup LCP$.

The agents in the set of witnesses, W, will be asked for reputation information about the target agent. If W or the initial set are empty then the agent will behave the same way as before the use of the Social Structure.

In the Art Agent Testbed, asking for reputation information has a cost. Using the social structure to select only agents who are more likely to have meaningful information could translate to a cost saving. Once the information is gathered, the opinion from the witness agent selected is weighted taking into account the social relationships linking the agent with the witnesses and the target agent.

This strategy has been integrated in one agent previously competing in the Art Competition. From an initial screening, we have noticed that many of the agents participating in the ART Competition, didn't implement the reputation module in their trust model. They only relied on direct interactions, because the total population in the competition was not large, allowing them to interact directly with almost every other agent. Therefore, the agent we selected is Simplet, whose trust and reputation model took into account witnesses information and made it easy to check the performance of the social structure. Simplet trust model is inspired by the LIAR [9] model. It has a reputation model built from an agent's own interactions, from its observations and from recommendations sent by other agents. It uses a violation detection process working with incomplete information. Internal rules are set to help in the detection of violation, such as rules finding contradictory information.

The next section will explain process used for the evaluation, explaining the parameters and the metrics used and will provide an analysis of the results.

## 4 Experiments and Analysis

One of the contributions of this paper is to provide a systematic evaluation of the use of the social structure, implemented as explained in the previous section, in a trust model.

Nine different configurations have been set up. The parameters of each configurations are shown in Table 1. The configurations contain parameters for the simulation enviroment, the Agent Art Testbed, and for the configuration of the social structure in SocialSimplet. For each configuration 50 runs are executed. All configurations have a constant set of agents consisting of those used by the 2008 Agent Art Competition, excluding Agent Uno [10], plus SocialSimplet. We felt that exlcuding Agent Uno was necessary because from an analysis of its trust model we noticed that it uses knowledge of the Agent Art Testbed to tune the parameters of its trust model. These parameters are known because they are explained in the documentation of the Agent Art Testbed, however they are not available for the agents competing to know. In our opinion, even if the parameters are available to the agent designers, they should be deliberately ignored for the sake of not biasing the trust model. Thus, including this agent might have affected the result of this evaluation.

The configurations can be grouped in three subcategories. The first category includes configurations A, B and C. They are the same settings used in the last Agent Art Testbed Competition [1]. Each of them has a different values for the parameters *#era to change* and *amount of expertise change*. These parameters determine the degree of dynamism in the games. By changing the level of expertise the agent has to react to variation in the other agents responses which might be due to different expertise rather than strategy.

The second category has a medium level of dynamism. It includes configurations D, E, F, G, J. Since the agent using the social structure makes a heavy use of reputation and expertise [2] requests, the costs of these requests may affect the final performance of the agent. This category of configurations show how the agent performs when getting this information is free. We test this condition

[1] From the Agent Art Testbed Web site.http://megatron.iiia.csic.es/art-testbed/competition2008.htm

[2] The expertise requests are called in the Art Competition setting *certainty request* because the agent communicates his level of certainty about an opinion on a particular era

with different values of total population, varying from 1 instance of each agent in the game to 10 instances for each agent.

The last category includes configurations H, I, L, M and N. In this category, we test the performance of the agent by varying parameters of the social structure (bottom Section of Table 1), such as the range of values used to identify the social relationship between two agents in the social structure. This set of configurations helped, as well, to identify optimal values for this range, in the ART Testbed. With the term optimal we mean values that allow the agent using the social structure to perform best.

| Parameters | Configurations | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | A | B | C | D | E | F | G | J | H | I | L | M | N |
| # games | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 |
| # timesteps | 90 | 90 | 90 | 90 | 90 | 90 | 90 | 200 | 90 | 90 | 90 | 90 | 90 |
| # eras | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| avg client/agent | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| client fee | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| opinion cost | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| certainty cost | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| reputation cost | 0.1 | 0.1 | 0.1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| # opinion msg | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| # certainty msg | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| deny self opinion | true | true | true | true | true | true | true | true | true | true | true | true | true |
| # eras to change | 1 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| amount expertise change | 0.05 | 0.1 | 0.3 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| # instance of each agent | 1 | 1 | 1 | 1 | 2 | 5 | 10 | 10 | 1 | 1 | 1 | 1 | 1 |
| initial reputation discard | true | true | true | true | true | true | true | true | false | true | true | true | true |
| expertise threshold | K | K | K | K | K | K | K | K | K | I | L | M | N |

K = 15.08, I = 2.08, L = 5.08, M = 10.08, N = 20.08

**Table 1.** Configurations settings for the evaluation. Top section contains settings for the Agent ART Testbed and the bottom section contains the settings concerning the social structure of SocialSimplet.

Several metrics are used to evaluate the performance. Each metric is observed for the Simplet and the corresponding agent using the social structure, which we will call SocialSimplet. The values are used to evaluate and compare the performance of the SocialSimplet with Simplet. The metrics considered are:

- the total utility, as in the ART Competition,
- the number of games won, grouped for each configuration settings,
- the number of fulfilled and violated interactions.

Every *interaction* is intended as an exchange of information between agents. This information can be an opinion about a painting or reputation information about another agent. These two types of information can be considered as the two services provided by the agents in the environment. This last metric is particular important in assessing the utility of the use of the social structure because it reveals how many times the trust model has been successful in selecting a trustworthy agent. To assess if an interaction has been fulfilled or violated, the

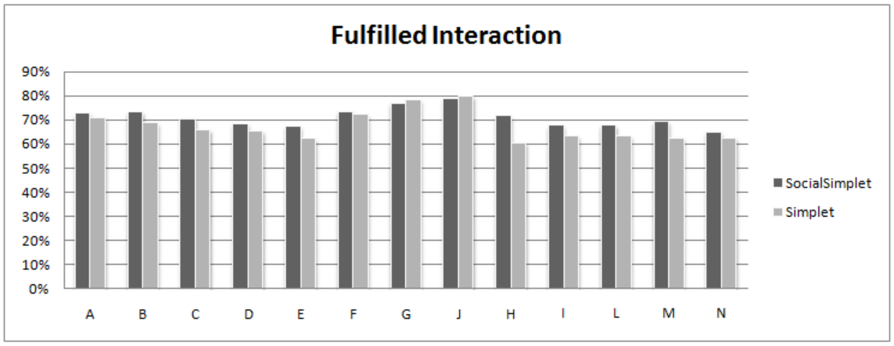


**Fig. 1.** % of Fulfilled Interaction for all configurations for Simplet and SocialSimplet.

agent uses the difference between the true value of the painting and the opinion provided by other agents in the previous interactions. The true value of each painting is revealed after every interaction, making this evaluation computable. The difference in the values, together with the expertise of the agent providing the opinion in the considered painting's era, allows the agent to decide if the interaction was fulfilled or violated. If the difference is above a certain threshold, the interaction is considered violated. This threshold is set in the internal trust model of the agent, therefore it is not considered as parameter of the experiments. However, since Simplet and SocialSimplet have the same trust model, this threshold will have the same value in both agents, meaning the comparison is not affected by this.

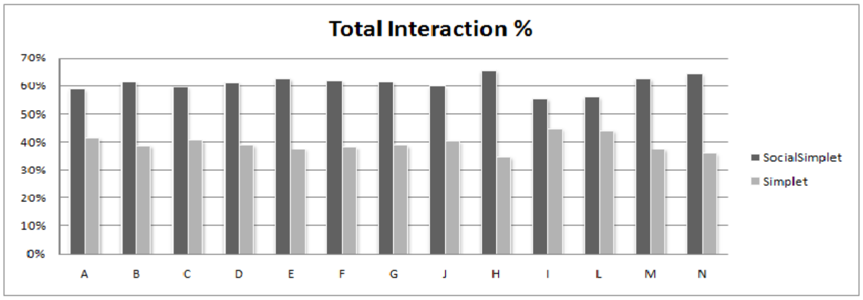


**Fig. 2.** % of Total Interaction for all configurations for Simplet and SocialSimplet.

Figure 1 shows the percentage of fulfilled interaction averaged over the fifty runs for each configuration. Although the chart shows that, overall, there is an improvement of only the 3.64%, the total number of interaction is considerably bigger in the SocialSimplet, as shown in the figure 2. This means that the effective number of fulfilled interactions is substantial. This improvement in the number of fulfilled interactions, clearly, shows that the internal trust model of the agent benefits from the information inferred from the social structure. This suggests that the trust model with the social structure can more accurately assess which agent is more likely to be trustworthy. Since the social structure is updated after every time step, SocialSimplet is only marginally affected by the change of dynamism in the configurations A, B and C, going from a 72.6% of fulfilled interactions in configuration A, to a 70.5% in configuration C. Simplet seems to suffer slightly more because of this change, loosing a bit more than 5% over the three set of games.

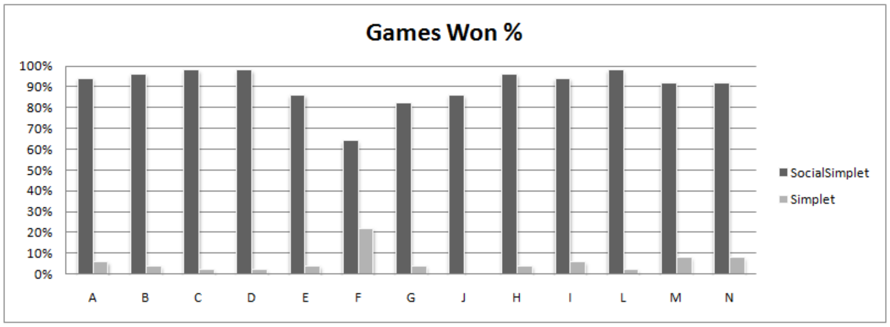


**Fig. 3.** % of Games Won for all configurations for Simplet and SocialSimplet.

Configurations F and G show the highest percentage of fulfilled interactions among the other configurations. This means that the social structure is particularly useful in those societies with a high number of agents, where direct interactions are not always possible with every agent in the society. In configuration J SocialSimplet nearly reaches a total of the 80% of fulfilled interactions against a total of more then 130 thousand interactions, hence having more than 102, 600 successful interactions. This shows also that the accuracy of the social structure improves over time, because after every interaction the social structure gains more information about the society . Therefore on very long games, such as the runs in configuration J, the agent can refine the social structure and obtain a better perception of the environment it is placed into. In configuration H we can observe that not discarding the initial reputation information does not seem to worsen the performance of Social Simplet, also because the social structure is updated at every time step, this allows the agent to correct any wrong assumption. From the set of games in configurations I, L, M and N, we can learn the optimal range of values of expertise distance seems to be in the range 10.081 to 15.081. These values are used to assess the types of social relationship linking the agents. We can also note that in configurations D, E, F and G, where the reputation and expertise requests have no cost, there is a progressive increment in the utility gained, in line with the increasing total population.

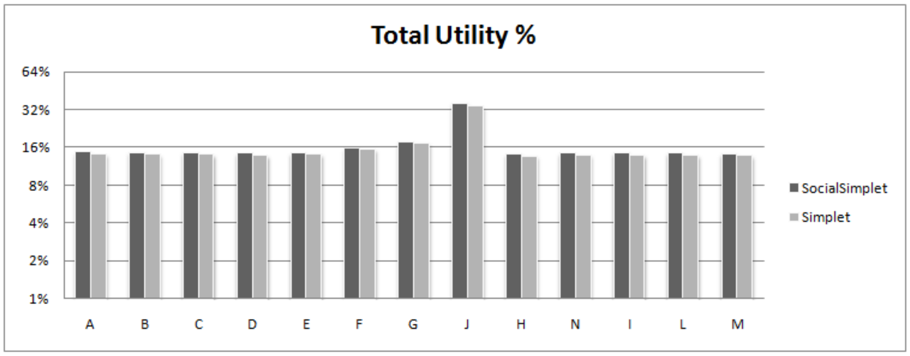


**Fig. 4.** % of Total Utility for all configurations for Simplet and SocialSimplet.

On the other hand, when considering the total utility, SocialSimplet performs better over all the different configurations; Figure 3 and Figure 4 show, respectively, the number of games in which SocialSimplet wins, and the margin by which it wins. This shows that SocialSimplet wins very often (almost 100% of the games), but by a very small margin. This is due to the client allocation share strategy, which is affected by the accuracy of the final appraisals produced. The agent chooses a total amount, representing the time taken to examine the painting, to be spent in generating its own opinion about a painting value. This amount affects the accuracy of the generated final appraisal; this parameter is a strategic decision on how the agent manages its money, and it is not affected by the changes made in SocialSimplet. In other words, the way Simplet administers its money is not part of the trust model, even if it is affected by it to an extent. The margin of improvement in accuracy of the appraisals produced is due to the better selections of other agents who are asked for their opinions, but it is harder to appraise given that the total utility is determined largely by the client allocation share strategy, which is the same for Simplet and SocialSimplet in this evaluation. Considering the only difference in the two models, Simplet and SocialSimplet, is the use of the social structure, it seems fair to infer that any improvement in the perfomance is due to the social structure.

These results demonstrate that the use of a social structure can offer benefits to agents. However, future work is needed to verify the effectiveness of this approach in different trust models and to improve the process of updating the social structure after every interaction, possibly including learning techniques.

## 5 Conclusion and Future Work

In this paper we showed how social structure analysis can be integrated into a trust model taking into account social relations among agents in the environment. Although the idea of a social structure has already been presented [2] in the past, there were no indication of how each agent would build this social network representation. The only attempt made is in [6]. However, the proposed model has never been implemented and validated. We presented a way for agents to build a social network representation of their local environment. Using interaction information such as reputation information, agents can maintain their own representation of such environments. With this extended perception of the environments, agents can make more informed decisions. For the first time, we have shown empirical evidence that a technique to build and maintain a social network representation of the environment allows a trust model to be more effective in selecting trustworthy agents.

This paper presents only the results of our initial investigation. Whilst the positive results have been encouraging, there is still much to explore. Future work includes a) further experiments to test the use of the social structure in each trust model, to verify, on a general scale, its effectiveness; b) exploration of methods for optimizing the maintenance and updating of the social structure, even with learning techniques; d) testing our model on a larger testbed.

## References

1. Axelrod, R.: The Evolution of Cooperation. Basic Books (1984)
2. Sabater, J., Sierra, C.: Reputation and social network analysis in multi-agent systems. In: AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems, New York, NY, USA, ACM (2002) 475–482
3. Fullam, K.K., Klos, T.B., Muller, G., Sabater-Mir, J., Topol, Z., Barber, K.S., Rosenschein, J., Vercouter, L.: The agent reputation and trust (art) testbed architecture. In: Proceeding of the 2005 conference on Artificial Intelligence Research and Development, Amsterdam, The Netherlands, The Netherlands, IOS Press (2005) 389–396
4. Krupa, Y., Hubner, J., Vercouter, L.: Extending the comparison efficiency of the ART testbed
5. Huynh, T.D., Jennings, N.R., Shadbolt, N.R.: An integrated trust and reputation model for open multi-agent systems. Autonomous Agents and Multi-Agent Systems **13**(2) (2006) 119–154
6. Ashri, R., Ramchurn, S.D., Sabater, J., Luck, M., Jennings, N.R.: Trust evaluation through relationship analysis. In: AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems, New York, NY, USA, ACM (2005) 1005–1011
7. Galaskiewicz, J., Wasserman, S.: Advances in social network analysis : research in the social and behavioral sciences / Stanley Wasserman, Joseph Galaskiewicz, editors. Sage Publications, Thousand Oaks, Calif. : (1994)
8. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1988)
9. Muller, G., Vercouter, L.: Decentralized monitoring of agent communications with a reputation model. In: Trusting Agents for Trusting Electronic Societies. (2004) 144–161
10. Murillo, J., Muñoz, V.: Agent uno: winner in the 2nd spanish art competition. Inteligencia artificial: Revista Iberoamericana de Inteligencia Artificial **12** (2008) 19–27

# Trust and Transitivity: a complex deceptive relationship

Rino Falcone and Cristiano Castelfranchi

National Research Council– Institute of Cognitive Sciences and Technologies
Via San Martino della Battaglia, 44  00185 - Roma, Italy
{rino.falcone, cristiano.castelfranchi}@istc.cnr.it

**Abstract.** Transitivity in trust is very often considered as a quite simple property, trivially inferable from the classical transitivity defined in mathematics, logic, or grammar. In fact the complexity of the trust notion suggests evaluating the relationships with the transitivity in a more adequate way. In this paper, starting from a socio-cognitive model of trust, we analyze the different aspects and conceptual frameworks involved in this relation and show how different interpretations of these concepts produce different solutions and definitions of trust transitivity.

**Keywords:** Trust, Transitivity, Degree of Trust, Task definition.

## 1 Introduction

Trust is becoming a research topic of major interest not only in Artificial Intelligence (AI) and in Multi-Agent Systems (MAS), but also more in general in Information and Communication Technologies (ICT). The main reason of this fact is that the more recent developments of the “interaction” paradigm of computation, are driving more and more towards the development of computational entities with a strong and well defined autonomy. These entities have to cooperate/conflict among them in conditions of open world for achieving their own goals.

In perspective, we are going towards an interaction scenario in which artificial entities and humans are indistinguishable from each other. In this view, the probability that we have to interact or cooperate with entities we do not have any personal experience with, will be growing, and the need of attributing *trustworthiness* to the potential partners becomes a fundamental prerequisite. And to model trust in the way in which humans have always done it, being them in the interaction loop, is particularly relevant.

Many different approaches and models of trust were developed in the last 15 years [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]: they contributed to clarify many aspects and problems about trust and trustworthiness, although many issues remain to be addressed and some elementary but not so trivial trust properties are left in a contradictory form.

One of them is the problem of trust transitivity. If ***X*** trusts ***Y***, and ***Y*** trusts ***Z***: What about the trust relationship between ***X*** and ***Z***? Different and sometimes diverging

answers were given to this problem. The question is not only theoretically relevant; it is very relevant form the practical point of view, for the reason we have just mentioned: acting in an open world, interacting with new people/agents.
In this paper we will present an analysis of the trust transitivity in the case in which a socio-cognitive model of trust is taken in consideration. Through this kind of model we are able to evaluate and partially cope with the complexity that the concept of transitivity introduces when applied to the trust relationship.

## 2 A socio-cognitive model of trust

In our socio-cognitive model of trust [11, 12, 13, 14] we consider trust as a layered notion, where the various more or less complex meanings are embedded one into the other. We analyzed the relevant relationships among these different layers and studied the possible transitions among them.
We developed the analysis of the *mental attitude* and *disposition* towards the trustee (considering beliefs like *evaluations* and *expectations*); the *intention* and *decision* based on the previous dispositions; the *act* of relying upon the trustee's expected behaviour; finally the *social interaction* and *relation* between trustor and trustee.

In particular we consider trust as a relational construct between the trustor (***X***), the trustee (***Y***), about a defined (more or less specialized) task ($\boldsymbol{\tau}$):

$$\boldsymbol{Trust\ (X\ Y\ C\ \tau\ g_X)}$$

where are also explicitly present both the ***X***'s goal ($\boldsymbol{g_X}$, respect to which trust is activated) and the role of the context (***C***) in which the relationship is going to happens. In fact, the successful performance of the task $\boldsymbol{\tau}$ will satisfy the goal $\boldsymbol{g_X}$.
So the ***X***'s mental ingredients of trust are: the goal $\boldsymbol{g_X}$, and a set of *main* beliefs:
$\boldsymbol{Bel\ (X\ Can_Y\ (\tau))}$
$\boldsymbol{Bel\ (X\ Will_Y\ (\tau))}$
$\boldsymbol{Bel\ (X\ ExtFact_Y\ (\tau))}$

where:
$\boldsymbol{Can_Y\ (\tau)}$ means that ***Y*** is potentially able to do $\boldsymbol{\tau}$ (in the sense that, under the given conditions, is competent, has the internal powers, skills, know-how, etc) (and this is believed by ***X)***;
$\boldsymbol{Will_Y\ (\tau)}$ means that, under the given conditions, ***Y*** potentially has the attributions for being willing, persistent, available, etc., on the task $\boldsymbol{\tau}$ (and this is believed by ***X)***;
$\boldsymbol{ExtFact_Y\ (\tau)}$ means that potentially there are a set of external conditions either favoring or hindering ***Y*** realizing the task $\boldsymbol{\tau}$ (and this is believed by ***X)***.

In our model we also consider that trust can be *graded*: ***X*** can have a *strong trust* that ***Y*** will realize the task (maybe 0.95 in the range (0,1)); or even just a *sufficient trust* that ***Y*** will achieve it (maybe 0.6 with a threshold of 0.55; and so on with other

possible values). For this we have introduced a quantification of the degree of trust ($\boldsymbol{DoT_{XY\tau}}$) and, in general, a threshold ($\boldsymbol{\sigma}$) to be overcome from this $\boldsymbol{DoT_{XY\tau}}$.
Given the previous analysis of the main components of the trust attitude ($\boldsymbol{g_X}$, ***Bel (X Can**$_Y$ **(τ)), Bel (X Will**$_Y$ **(τ)), Bel (X ExtFact**$_Y$ **(τ)))***, we can say that this degree is, on its turn, resultant from the several *quantifications* of these components:

$$\boldsymbol{DoT_{XY\tau} = f(DoC_X\,(Opp_Y(\tau)), DoC_X\,(Ability_Y(\tau)), DoC_X\,(Willingness_Y(\tau)))}$$

where: ***f*** is in general a function that preserves monotonicity;
$\boldsymbol{DoC_X\ (Opp_Y(\tau))}$ is the ***X***'s degree of credibility about the external opportunities (positively or negatively) interfering with ***Y***'s action in realizing the task $\boldsymbol{\tau}$;
$\boldsymbol{DoC_X\ (Ability_Y(\tau))}$ is the ***X***'s degree of credibility about the ***Y***'s ability/competence to perform $\boldsymbol{\tau}$;
$\boldsymbol{DoC_X\ (Willingness_Y(\tau))}$ is the ***X***'s degree of credibility about the ***Y***'s willingness to perform $\boldsymbol{\tau}$.

We are ignoring the subjective certainty of the pertinent beliefs (how much sure is ***X*** of its evalutative beliefs about that specific ***Y***'s quality, that is a meta-belief; in fact we can say that this factor is integrated with the other). At the same time we are ignoring for now the value of the goal ($\boldsymbol{g_X}$).

So trivially ***X*** will trust ***Y*** about the task $\boldsymbol{\tau}$ if

$\boldsymbol{DoT_{XY\tau} > \sigma}$

that means that a set of analogous conditions must be realized about the other quantitative elements ($\boldsymbol{DoC_X\ (Opp_Y(\tau)), DoC_X\ (Ability_Y(\tau)), DoC_X\ (Willingness_Y(\tau))}$). We do not consider in this paper the detailed analysis of how the degree of trust is resulting by the more elementary components and we also omit of considering the potential positive and negative interferences among the components themselves.

## 3 Transitivity in Trust

Many authors have questioned whether the transitive property can be applied to trust. In more specific words many of them have presented this problem in this way:
If ***X*** trusts ***Y***, and ***Y*** trusts ***Z***: What about the trust relationship between ***X*** and ***Z***?
Their answers are different and very briefly we will analyze some of them in §4.

We are now interested to translate this problem in our terms of trust.

First of all, we do not consider the unspecified case "***X*** trusts ***Y***" because in our model an agent has to trust another agent with respect a task (either very well defined or less defined and abstract); this task directly derives from the goal the trustor has to reach with the trust attribution. So we have to transform "***X*** trusts ***Y***" in "***X*** trusts ***Y*** about $\boldsymbol{\tau}$". And given the graded qualification of trust we have that:

$\boldsymbol{DoT_{XY\tau} > \sigma}$

this means in particular that ***X*** *believes* that ***Y*** is potentially *able* and *willing* to do $\boldsymbol{\tau}$ and that the external conditions in which ***Y*** will perform its task are at least not so opposite to the task realization (may be also they are neutral or favorable).

So this ***Y***'s trustworthiness (perceived/believed by ***X***) is based on these specific beliefs.
At the same way "***Y*** trusts ***Z***" becomes "***Y*** trusts ***Z*** about $\boldsymbol{\tau_1}$" (about the difference between $\boldsymbol{\tau}$ and $\boldsymbol{\tau_1}$, see later) with the same particular ***Y***'s beliefs about ***Z*** and the external conditions.
Also in this case we can say that there is a threshold to be overcome and the condition:

$\boldsymbol{DoT_{YZ\tau 1} > \sigma_1}$
successfully satisfied in case of trust attribution.

If we have to consider the trust relationship between "***X*** and ***Z***" as a consequence of the previous trust relationships between "***X*** and ***Y***" and between "***Y*** and ***Z***" we have to define the task on which this relationship should be based (question of *assimilation* between $\boldsymbol{\tau}$ and $\boldsymbol{\tau_1}$) and the degree of trust that must be overcome even from ***X***:

$\boldsymbol{DoT_{XZ\tau} > \sigma_2}$
with the consideration of the threshold $\boldsymbol{\sigma_2}$.

The role of the trust threshold is quite complex and can have an overlapping with the ingredients of trust. We strongly simplify in this case considering $\boldsymbol{\sigma}$ as dependent just from the specific intrinsic characteristics of the trustor (those that define an agent intrinsically: prudent, reckless, and so on) independently from the external circumstances (on the contrary, these factors affect the degree of trust, by affecting the more elementary beliefs above showed).
So, we can say that in this approximation (for the same agent the trust threshold is always the same):
$\boldsymbol{\sigma = \sigma_2}$

In the case in which all the agents are defined as having the same intrinsic characteristics (this fact is possible in the case of artificial entities), we can also say that:
$\boldsymbol{\sigma = \sigma_1 = \sigma_2}$

Moreover, as we just saw, not less important in our approach is that trust is an expectation and a bet *grounded on and justified by* certain beliefs about ***Y***. ***X*** trusts ***Y*** on the basis of the evaluation of ***Y***'s "virtues/qualities", not just on the basis of a statistical sampling, some probability.
The *evaluation* about the needed "qualities" of ***Y*** for $\boldsymbol{\tau}$ are the *mediator* for the decision to trust ***Y***. This mediation role is fundamental also in trust transitivity.

Let us now consider the case of the differences between the tasks in the different relationships.

For the trust transitivity the two tasks should be the same ($\boldsymbol{\tau} = \boldsymbol{\tau_1}$). Is this equality enough?
Suppose for example that there are 3 agents: John, Mary and Peter; and suppose that John trusts Mary about "organizing scientific meetings" (task $\boldsymbol{\tau}$), at the same time Mary trusts Peter about "organizing scientific meetings" (again task $\boldsymbol{\tau}$). Can we deduce that, given the transitivity of trust: John trusts Peter about "organizing scientific meetings"? Is in fact transferable that task evaluation? Given the trust model defined in §2 the situation is more complex and there are possible pitfalls lurking: Mary is the central node for that trust transfer and she plays different roles (and functions) in the first case (when her trustworthiness is about *to realize* the task $\boldsymbol{\tau}$, and in the second case (when her trustworthiness is about *evaluating* the Peter's trustworthiness on the task $\boldsymbol{\tau}$).

The situation is even clearer if we split in the example the two kinds of competences: ***X*** trusts very much ***Y*** as medical doctor; ***X*** knows that ***Y*** trusts ***Z*** as mechanic; will ***X*** trust ***Z*** as mechanic? Not necessarily at all: if ***X*** believes that ***Y*** is a good evaluator of mechanics he will trust ***Z***; but, if ***X*** believes that ***Y*** is a very naive in this domain, and is frequently swindled by people, he will not trust ***Z***. In the previous example the transition looks more plausible, natural, just because the task $\boldsymbol{\tau}$ is the same, and it is reasonable (but not necessary) that if ***Y*** is very skilled and competent in task $\boldsymbol{\tau}$, she will also be a good evaluator of other people on the same task.

So for considering transitivity of trust as a valid property (in the classical way in which it is defined) in these types of situations, we have to *assimilate* the task with the evaluation of that task itself:
$\boldsymbol{Bel\ (X\ (Trustworthiness_Y(\tau)\ implies\ Trustworthiness_Y(evaluation\text{-}of\ (\tau))))}$
In words, ***X*** has to believe that if ***Y*** is trustworthy on the task $\boldsymbol{\tau}$, it is also trustworthy on the meta-task of evaluating $\boldsymbol{\tau}$ (in both the situations the ***X***'s mental ingredients defining the trust in ***Y*** allow to overcome the threshold).

We do not consider in this paper the truthfulness of this hypothesis (and the consequent properties both in the more elementary mental ingredients of the interacting actors, and in the tasks' features): in fact, our trust model is apt to analyze in detail this problem. However, we want underline the difference of the involved tasks in the relationships and the necessity of taking into consideration these differences before generally speaking of trust transitivity.

So resuming we have the *more basic case* of the relationship between *trust* and *transitivity* so defined:
***if***

i) "***X*** trusts ***Y*** about $\boldsymbol{\tau}$" (that means: $\boldsymbol{DoT_{XY\tau} > \sigma_X}$) and

ii) "***Y*** trusts ***Z*** about $\boldsymbol{\tau}$" (that means: $\boldsymbol{DoT_{YZ\tau} > \sigma_Y}$) and

iii) "Bel(***X*** (***Y*** trusts ***Z*** about $\boldsymbol{\tau}$))" and

iv) "$\sigma=\sigma_X=\sigma_Y$" (the trust threshold is the same for each agent and for each relation) and
v) "***Bel (X (Trustworthiness**$_Y$**(τ) implies Trustworthiness**$_Y$**(evaluation-of**$_Y$ **(τ))))***" (***Y*** is equally trustworthy in the realization of the task and in evaluating others performing that task)
***then***
vi) "***X*** trusts ***Z*** about $\tau$" (that means: $DoT_{XZ\tau} > \sigma$)

The fact that are true: $DoT_{XY\tau} > \sigma$ and $DoT_{YZ\tau} > \sigma$ and $DoT_{XZ\tau} > \sigma$

does not mean that necessarily $DoT_{XY\tau} = DoT_{YZ\tau} = DoT_{XZ\tau}$. As we have seen in the §2 these degrees are dependent from the internal beliefs on the different components, and they are resulting from different sources not necessarily all common to the involved agents.

### 3.1 Trust and Transitivity in the delegated subtasks

Another interesting case of the relationship between trust and transitivity is when we have the following situation:
"***X*** trusts ***Y*** about $\tau$" and, in realizing the task ($\tau$), ***Y*** delegates parts of the task itself to other agents ***Z***, ***W*** (for example the delegated subtasks are respectively $\tau_1$ and $\tau_2$).
Then if we suppose that ***X*** is aware of this delegation; what we can say (on the basis of the trust relationship between ***X*** and ***Y***) about the trust between ***X*** and ***Z*** with respect to $\tau_1$? And between ***X*** and ***W*** with respect $\tau_2$?

Suppose, for example, that John trusts Mary about "organizing a scientific meeting" and that Mary delegates Peter to "organize the registration process", and delegates Alice to "organize the sponsoring of the event". What about the John's direct trust on Peter (as organizing the registration process) and the John's direct trust on Alice (as organizing the sponsoring of the event)? Are these trust relationships the same of the Mary's ones? How are they *mediated* by the John's trust on Mary?

We have that $DoT_{John,Mary,\tau} > \sigma$ and $DoT_{Mary,Peter,\tau 1} > \sigma$ and $DoT_{Mary,Alice,\tau 2} > \sigma$
But from these assumptions does not necessarily follow that:

$DoT_{John,Peter,\tau 1} > \sigma$ and $DoT_{John,Alice,\tau 2} > \sigma$

In fact, John should know how exactly the delegation of the subtasks is realized and on what basis is founded. In a trust relationship, as we have seen in §2, are involved not only qualities about abilities and skills but also qualities about willingness, availability, and so on. So these others qualities could be elicited by the specific relation with the trustor (delegating agent) and in some cases are strictly related with the interaction history among the agents (see §3.2 for a more detailed analysis of this). In these cases is more difficult for John to evaluate which could be the Peter's and Alice's performances (and then their trustworthiness).

In addition may be there is a particular interaction among the subtasks and the main task in which Mary plays a specific role of integration and substitution (in presence of shortcomings of the other agents) that is essential for the success of the complete task. Also in this case John trusting Peter and Alice, at the same way of Mary, should be aware to be able of playing the same role.

So we can say that for applying the trust transitivity to the cases of subtasks delegation, we have to analyze in deep detail: on the one hand the set relationships among task and its subtasks (and how they are distributed among the agents in play), and, on the other hand, how the executing agents are *motivated* and *activated* in the task realization by the relationship with the trustor.

Resuming in the case of delegated subtasks we can say that:
***if***
i) "$\boldsymbol{X}$ trusts $\boldsymbol{Y}$ about $\boldsymbol{\tau}$" (that means: $\boldsymbol{DoT_{XY\tau} > \sigma_X}$) and
ii) "$\boldsymbol{Y}$ trusts $\boldsymbol{Z}$ about $\boldsymbol{\tau_1}$" (that means: $\boldsymbol{DoT_{YZ\tau 1} > \sigma_Y}$) and
iii) "Bel($\boldsymbol{X}$ ($\boldsymbol{Y}$ trusts $\boldsymbol{Z}$ about $\boldsymbol{\tau_1}$))" and
iv) "$\boldsymbol{Y}$ trusts $\boldsymbol{W}$ about $\boldsymbol{\tau_2}$" (that means: $\boldsymbol{DoT_{YZ\tau 2} > \sigma_Y}$) and
v) "Bel($\boldsymbol{X}$ ($\boldsymbol{Y}$ trusts $\boldsymbol{W}$ about $\boldsymbol{\tau_2}$))" and
vi) $\boldsymbol{\tau_1}$ is a subtask of $\boldsymbol{\tau}$ (realizing $\boldsymbol{\tau_1}$ are realized part of the results of $\boldsymbol{\tau}$) and
vii) $\boldsymbol{\tau_2}$ is a subtask of $\boldsymbol{\tau}$ (realizing $\boldsymbol{\tau_2}$ are realized part of the results of $\boldsymbol{\tau}$) and
viii) "$\boldsymbol{\sigma=\sigma_X=\sigma_Y}$" (the trust threshold is the same for each agent and for each relation) and
ix) $\boldsymbol{X}$ believes that the interactional history between $\boldsymbol{Y}$ and $\boldsymbol{Z}$ is not essential for the $\boldsymbol{Z}$'s performance about $\boldsymbol{\tau_1}$ and
x) $\boldsymbol{X}$ believes that it is able to integrate (as well as $\boldsymbol{Y}$) the $\boldsymbol{Z}$'s deficiencies in realizing $\boldsymbol{\tau_1}$ and
xi) $\boldsymbol{X}$ believes that the interactional history between $\boldsymbol{Y}$ and $\boldsymbol{W}$ is not essential for the $\boldsymbol{W}$'s performance about $\boldsymbol{\tau_2}$ and
xii) $\boldsymbol{X}$ believes that it is able to integrate (as well as $\boldsymbol{Y}$) the $\boldsymbol{W}$'s deficiencies in realizing $\boldsymbol{\tau_2}$ and
***then***
xiii) "$\boldsymbol{X}$ trusts $\boldsymbol{Z}$ about $\boldsymbol{\tau_1}$" (that means: $\boldsymbol{DoT_{XZ\tau 1} > \sigma}$) and
xiv) "$\boldsymbol{X}$ trusts $\boldsymbol{W}$ about $\boldsymbol{\tau_2}$" (that means: $\boldsymbol{DoT_{XW\tau 2} > \sigma}$).

### 3.2 Competence and Willingness in Transitivity

The need for a careful qualitative consideration of the nature of the link between the trustor and the trustee, is even more serious.

Not only it is fundamental (as we have argued) to make explicit and do not forget the specific "task" (activity, and thus "qualities") $\boldsymbol{X}$ is trusting $\boldsymbol{Y}$ or $\boldsymbol{Z}$ about, but it is even necessary to consider the different dimensions/components of the trust disposition (evaluation), decision, and relation.

In our model (a part from the basic though and feeling that I have not to worry about

***Y***, that there is no danger from ***Y***'s side, that I do not need diffidence and a defensive attitude), trust has two basic nucleuses:
(i) ***Y***'s *competence*, ability, for correctly performing the delegated task;
(ii) ***Y***'s *willingness* to do it, to act as expected.

The two dimensions (and 'virtues' of ***Y***) are quite independent on each other: ***Y*** might be very well disposed and willing to do, but not very competent or unable; ***Y*** might be very expert and skilled, but not very reliable: unstable, unpredictable, not well disposed, insincere, dishonest, etc.

Now, this (at least) double dimensions affect transitivity. In fact, even assuming that the *competence* is rather stable (see below) (and that ***Y*** is a good evaluator of ***Z***'s *competence*) not necessarily ***Z***'s *willingness* is equally stable and transferable from ***Y*** to ***X***. This is a more relation-based dimension.
***Y*** was evaluating ***Z***'s *willingness* to do as expected on the basis of their specific *relation*. Is ***Z*** a friend of ***Y***? Is there a specific benevolence, or values sharing, or gratitude and reciprocity, or obligation and hierarchical relation, etc.? Not necessarily the reasons that ***Z*** would have for satisfying ***Y***'s expectation and delegation would be present (or equally important) towards ***X***. ***X***'s relation with ***Z*** might be very different. Are the reasons/motives motivating ***Z*** towards ***Y***, and making him reliable, transferable or equally present towards ***X***? Only in this case it would be reasonable for ***X*** to adopt ***Y***'s trustful attitude and decision towards ***Z***.
Only certain kinds of relations will be generalized from ***Y*** to ***X***; for example, if ***Y*** trusts ***Z*** only because it is an economic exchange, only for ***Z***'s interest in money, reasonably ***X*** can become a new client of ***Z***; or if ***Y*** relies on ***Z*** because ***Z*** is a charitable person, generously helping (without any prejudice and discrimination) poor suffering people, and ***X*** is in the same condition of ***Y***, than also ***X*** can trust in ***Z***.
In sum,

- ***Y***'s expectation about ***Z***'s reliable behavior, in particular about ***Z***'s "adoption" of ***Y***'s goal (help, etc.) *depends on the relationship between **Y** and **Z***, and in particular on ***Z****'s attitude towards* ***Y*** (and reasons for goal-adoption); if the *relation* between ***X*** and ***Z***, and in particular ***Z***'s attitude towards ***X*** (and reasons for goal-adoption), would be analogous, then the trust "transfer" would be reasonable.

### 3.3 Trust Dynamics affects Transitivity

Moreover, we have shown ([13], [14]) that ***Z***'s willingness, and even ability, can be affected, increased, by ***Y***'s trust and reliance (this can affect ***Z***'s commitment, pride, effort,... attention, study,...). ***Z***'s *trustworthiness* is improved by ***Y***'s trust and delegation. And ***Y*** might predict and calculate this in her decision to rely on ***Z***.
However, not necessarily the effect of ***Y*** on ***Z***'s trustworthiness will be produced also by another trustor. Thus, also this will affect "transitivity": suppose that ***Y***'s trust and delegation to ***Z*** makes him more trustworthy, improves ***Z***'s willingness or ability (and ***Y*** trusts and relies on ***Z*** on the basis of such expectation); not necessarily ***X***'s reliance on ***Z*** would have the same effect. Thus even if ***X*** knows that ***Y*** reasonably trusts ***Z*** (for something) and that he is a good evaluator and decision-maker, not necessarily ***X*** can

have the same trust in ***Z***, since perhaps ***Z***'s trustworthiness would not be equally improved by ***X***'s reliance.
Trust 'dynamics' between ***Y*** and ***Z***, is not automatically identical to trust "dynamics" between ***X*** and ***Z***.

### 3.4 The paradoxical self-transitivity of Trust

Our analytical model of trust components and steps allows also a nice reading of trust formulation (and then decision) in some sort of *internal transitivity*.
Actually, when ***X*** arrives to formulate a positive trust judgment of ***X*** itself about ***Y***, ***X*** must necessarily trust ***X***-self as a good evaluator; ***X*** has to find reliable its formulation of the *evaluation* about ***Y*** (that for us is a necessary first component of trust attitude and decision).

Thus in the case of self-transitivity, we can say that:
***if***
(i) ***X*** has a good *evaluation* about ***Y*** relative to $\boldsymbol{\tau}$ and
(ii) ***X*** (implicitly or explicitly) trusts ***X*** as for evaluating ***Y*** relatively to $\boldsymbol{\tau}$
***then***
(iii) "***X*** trusts ***Y*** about $\boldsymbol{\tau}$"

If (on the basis of previous failures) I cannot trust my-self as a good evaluator of people in a given domain, although I would formulate a good judgment of ***Y*** and a good prediction, my experience says that I cannot rely on my evaluation, and thus on ***Y***. The situation is not so different from real transitivity: when I get ***Y***'s evaluative judgment about ***Z*** and I have to decide whether to trust ***Z*** or not, on that basis. First, I have to rely on ***Y***'s evaluation (that is on ***Y*** as evaluator).
The only difference is that *usually* I trust my judgment automatically and by-default, without any explicit reasoning; while *usually* I reflect about trusting or not ***Y*** as a source of evaluations. But also this is not general: frequently also trusting ***Y*** in her/his evaluation is or becomes rather non-reflected, automatic: I have learned that ***Y***'s judgment are reliable, and I automatically "adopt" them; or I have learned that in this context C or that class of people C is reliable, and then I trust ***Y***'s judgment, since ***Y*** belongs to C.

## 4 Transitivity and Trust: Related Work

The necessity of modeling trust in the social networks is becoming more and more important, and a set of new definitions are emerging in different domains of computing: *computational trust*, *trust propagation*, *trust metrics*, *trust in web-based social networks*, and so on. Most of these concepts are strictly linked with the goal of inferring trust relationships between individuals that are not directly connected in the networks. For this reason the concept of trust transitivity is very often considered and used in these approaches.

A relevant example is given from the Josang's approach; he introduces the subjective logics (an attempt of overcoming the limits of the classical logics) for taking in consideration the uncertainty, the ignorance and the subjective characteristics of the beliefs [15]. Using this approach Josang addressed the problem of trust transitivity in different works [16], till the last developments, see [17], where it is recognized the intrinsic cognitive nature of this phenomenon. However, the main limits of this approach are that trust is in fact the *trust in the information sources*; and the transitivity regards two different tasks (referred to our formalism: $\boldsymbol{\tau} \neq \boldsymbol{\tau_1}$: ***X*** has to trust the evaluation of ***Y*** (task $\boldsymbol{\tau}$) with respect ***Z*** as realizing another task (task $\boldsymbol{\tau_1}$, for example as mechanic). As we showed before, this difference is really relevant for the transitivity phenomenon. In addition, also the first task (***Y*** as evaluator) is just analyzed with respect to the *property of sincerity* (and this is a confirmation of the constrained view of trust phenomenon; they write: "*A*'s disbelief in the recommending agent *B* means that *A* thinks that *B* consistently *recommends the opposite* of his real opinion about the truth value of *x*"; where *A*, *B*, and *x* are, in our terms, respectively ***X***, ***Y*** and $\boldsymbol{\tau_1}$). But in trusting someone as evaluator of another agent (with respect to a specific task), I have also to consider his *competence* as evaluator, not just his *sincerity*. Trust is based on ascribed qualities. ***Y*** could be completely sincere but at the same time completely inappropriate to evaluate that task. So the limits of this approach of a adequate treatment of the trust transitivity are quite clear.

Many other authors [18, 19], developed algorithms for inferring trust among agents not directly connected. These algorithms differ from each other in the way they compute trust values and propagate those values in the networks. In any case when trust transitivity is introduced, this phenomenon is not analyzed with respect the complexity it contains and that we tried to explain in the previous paragraphs.

## 5 Conclusions

The relationships between Transitivity and Trust (like transitivity in partial order and equivalence relations) represent an important element to well-understand the intimate nature of the trust concept. Since trust is considered the glue of social interactions, its complex nature has to be deeply investigated if we want realize a careful and precise model to be transfered in the artificial societies. The analysis of the trust properties is, in this sense, a very useful tool. And transitivity (among the other properties) is one of the more interesting one. In this paper we analyzed the relationships between trust and transitivity, showing in particular, how this analysis implicates the evaluation of the tasks involved in the different relationships, of the qualities of the agents in play in those specific tasks and of the particular relationships among the agents (and of their interaction histories and contexts).

We tried to show how a too trivial simplification of the transitive property when applied to the trust concept leads to wrong conclusions about the model of the phenomenon we are studying.

# References

[1] Marsh, S.P., (1994), Formalising Trust as a computational concept. PhD thesis, University of Stirling. Available at: http://www.nr.no/abie/papers/TR133.pdf.

[2] Jonker, C., and Treur, J., (1999), Formal Analysis of Models for the Dynamics of Trust based on Experiences, Autonomous Agents '99 Workshop on "Deception, Fraud and Trust in Agent Societies", Seattle, USA, May 1, pp.81-94.

[3] Barber, S., and Kim, J., (2000), Belief Revision Process based on trust: agents evaluating reputation of information sources, *Autonomous Agents 2000 Workshop on "Deception, Fraud and Trust in Agent Societies",* Barcelona, Spain, June 4, pp.15-26.

[4] Jones, A.J.I., Firozabadi, B.S., (2001), On the characterization of a trusting agent: Aspects of a formal approach. In Castelfranchi, C., Tan, Y.H., (Eds.), Trust and Deception in Virtual Societies. Pp. 55-90. Kluwer, Dordrecht.

[5] Resnick P., Zeckhauser, R., (2002), Trust among strangers in internet transactions: Empirical analysis of eBay's reputation systems. In Baye, R. (Editor), The economico of the internet and e-commerce. Vol. 11 of Advances in Applied Microneconomics. Elsevier Science.

[6] Yu, B., Singh, M.P., (2003), Searching social networks. In Proceedings of the second international joint conference on autonomous agents and multi-agent systems (AAMAS). Pp. 65-72. ACM Press.

[7] Singh, M.P., (2003), Trustworthy Service Composition: challenger and Research Questions. in Falcone, R., Barber, S., Korba, L., Singh, M., (Eds.), Trust Reputation, and Security: Theories and Practice. Lecture Notes on Artificial Intelligence, n°2631, Springer.

[8] Sabater, J. (2003), Trust and Reputation for Agent Societies, PhD thesis, Universitat Autonoma de Barcelona.

[9] Hang, C.W., Wang, Y., Singh, M.P., (2009), Operators for Propagating Trust and their Evaluation in Socila Networks. In Proceedings of the eight international joint conference on autonomous agents and multi-agent systems (AAMAS).

[10] Ziegler, C.N., (2009), On Propagation Interpersonal Trust in Social Network. In Golbeck, J., (Ed.), Computing with Social Trust. Human Computer Interaction Series, Springer.

[11] Castelfranchi C., Falcone R., (1998) Principles of trust for MAS: cognitive anatomy, social importance, and quantification, *Proceedings of the International Conference of Multi-Agent Systems (ICMAS'98*), pp. 72-79, Paris, July.

[12] Falcone R., Castelfranchi C., (2001). Social Trust: A Cognitive Approach, in *Trust and Deception in Virtual Societies* by Castelfranchi C. and Yao-Hua Tan (eds), Kluwer Academic Publishers, pp. 55-90.

[13] Falcone R., Castelfranchi C. (2001), The socio-cognitive dynamics of trust: does trust create trust? In *Trust in Cyber-societies: Integrating the Human and Artificial Perspectives* R. Falcone, M. Singh, and Y. Tan (Eds.), LNAI 2246 Springer. pp. 55-72.

[14] Castelfranchi C., Falcone R., (2010), Trust Theory: A Socio-Cognitive and Computational Model, John Wiley and Sons, (in press).

[15] Josang A., A logic for uncertain probabilities, International journal of Uncertain, Fuzziness and Knowledge-based Systems, 9(3):279-311, June, 2001.

[16] Josang A., Gray E., and Kinateder M., Simplification and Analysis of Transitive Trust Networks, Web Intelligence and Agent Systems, 4(2): 139-161, 2006.

[17] Bhuiyan T., Josang A., Xu Y., An analysis of trust transitività taking base rate into account, In: proceeding of the Sixth International Conference on Ubiquitous Intelligence and Computing, 7-9 July 2009, University of Queensland, Brisbane, 2009.

[18] Li, X., Han, Z., Shen, C., Transitive trust to executables generated during runtime, second International Conference on Innovative Computing, Information and Control, 2007.

[19] Golbeck J., Hendler J., Inferring binary trust relationships in web-based social networks, ACM Transactions on Internet Technology, 6(4), 497-529, 2006.

# Optimizing Advisor Network Size in a Personalized Trust-Modelling Framework for Multi-Agent Systems

Joshua Gorner and Robin Cohen

David R. Cheriton School of Computer Science, University of Waterloo
Waterloo, Ontario, N2L 3G1
{jgorner, rcohen}@uwaterloo.ca

**Abstract.** This paper explores potential improvements to Zhang's personalized trust approach for e-commerce, in particular examining means of optimizing the number of advisors that each buyer maintains in their social network. We propose three such improvements, two directly relating to the size of the network (through either the use of a threshold or by setting a maximum network size), and a third which may indirectly reduce the necessary size by ensuring that relevant trust information from users outside this network remains available. We provide examples to illustrate these approaches, and propose future work to optimize certain aspects of these methods and evaluate their overall effectiveness.

**Keywords:** user modelling, multi-agent systems, trust modelling, social networks, electronic commerce

## 1 Introduction

Zhang [1] has recently proposed a novel trust-based framework primarily developed for use in agent-oriented electronic commerce. This system relies on a model of the trustworthiness of a particular buyer's advisors — that is, other buyers within the system that report on sellers — which incorporates estimates of the advisor's private and public reputations. Users create a social network of trusted advisors, and sellers will offer better rewards in order to satisfy trusted advisors and thus build their own reputations.

In this paper, we look at one of the open questions regarding the optimal size of a user's advisor network. Retaining a large network could be inefficient, and may result in reduced accuracy in determining trust in a seller: many potential advisors will not have similar tastes to the current user, so a large network could make predictions less accurate. However, with a smaller network, there is a greater risk that the advisors will have insufficient experience [2]. We suggest three modifications to Zhang's model which may help in this regard, and demonstrate how the revised approach would operate in each case.

## 2 Related Work in Selecting Advisors

### 2.1 Mechanism

The mechanism we examine here [1] uses a multi-stage "personalized" approach for representing reputation in an e-commerce system, summarized below.

We note at the outset that the model only considers two possible ratings, positive (1) or negative (0). A buyer, denoted by $b$, first constructs a measure of private reputation of advisors, based on the advisors' ratings for sellers that $b$ has previously dealt with, and representing an estimation of the probability that an advisor $a$ will give fair ratings to $b$. Each pair of ratings considered is weighted based on the amount of time that separates the submission of the two ratings using a "forgetting factor", $\lambda$ $(0 \leq \lambda \leq 1)$, such that a pair of ratings will have a greater weight if they are made within close time proximity. This measure is known as "private" reputation, since this evaluation makes use of the buyer's own experiences, and is calculated as shown in equation 1:

$$\alpha = N_p + 1, \qquad \beta = N_{all} - N_p + 1, \qquad R_{pri}(a) = E(Pr(a)) = \frac{\alpha}{\alpha + \beta} \tag{1}$$

In Equation 1, $N_p$ represents the sum of the weights of all positive rating pairs (that is, pairs of identical ratings) for all sellers commonly rated by $b$ and $a$, and $N_{all}$ is the total sum of weights of all rating pairs involving $b$ and $a$. If $\lambda = 0$, then $N_p$ and $N_{all}$ will be simply the counts of the applicable types of rating pairs.

Next, the public reputation of an advisor, or the probability that an advisor will provide "consistent" ratings, is calculated similarly, using equation 2:

$$\alpha' = N_c + 1, \qquad \beta' = N'_{all} - N_c + 1, \qquad R_{pub}(a) = \frac{\alpha'}{\alpha' + \beta'} \tag{2}$$

Here, $N_c$ represents the number of ratings, provided by an advisor $a$, that are consistent with the majority of ratings provided for that seller up to the moment that this additional rating is submitted, while $N'_{all}$ is the total number of ratings provided by $a$.

At this point, given some maximum acceptable level of error $\epsilon \in (0, 1)$ and level of confidence $\gamma \in (0, 1)$, we derive $w$, the reliability of the private reputation value, which we then use in our calculation of the overall trustworthiness of $a$. As can be seen, a more reliable private reputation will have a greater effect on the overall result:

$$N_{min} = -\frac{1}{2\epsilon^2} ln \frac{1-\gamma}{2} \tag{3}$$

$$w = \begin{cases} \frac{N_{all}}{N_{min}} & \text{if } N_{all} < N_{min} \\ 1 & \text{otherwise} \end{cases} \tag{4}$$

$$Tr(a) = wR_{pri}(a) + (1 - w)R_{pub}(a) \tag{5}$$

Once this value has been calculated for each advisor, a similar approach can be taken for the trustworthiness of a given seller $s$. First the buyer $b$ calculates her private reputation of $s$, or the probability that $s$ will provide good service, based on $b$'s past experiences with $s$. This makes use of the number of positive ratings, $N^b_{pos,i}$, and negative ratings, $N^b_{neg,i}$, she provided for $s$ in each time window $T_i$, as well as the forgetting factor $\lambda$:

$$R_{pri}(s) = \frac{\sum_{i=1}^{n} N^b_{pos,i}\lambda^{i-1} + 1}{\sum_{i=1}^{n}(N^b_{pos,i} + N^b_{neg,i})\lambda^{i-1} + 2} \quad (6)$$

Next we derive the public reputation of the seller, the probability that the seller will provide good service given all advisors' past experiences with $s$, taking into account $b$'s own model of trustworthiness of each advisor $a_j$. We first make use of the following discounting functions to determine $b$'s trust of ratings provided by each $a_j$:

$$D^{a_j}_{pos_i} = \frac{2Tr(a_j)N^{a_j}_{pos,i}}{(1 - Tr(a_j))(N^{a_j}_{pos,i} + N^{a_j}_{neg,i}) + 2}, D^{a_j}_{neg_i} = \frac{2Tr(a_j)N^{a_j}_{neg,i}}{(1 - Tr(a_j))(N^{a_j}_{pos,i} + N^{a_j}_{neg,i}) + 2} \quad (7)$$

The public reputation of $s$ is itself calculated as follows:

$$R_{pub}(s) = \frac{[\sum_{j=1}^{k}\sum_{i=1}^{n} D^{a_j}_{pos,i}\lambda^{i-1}] + 1}{[\sum_{j=1}^{k}\sum_{i=1}^{n}(D^{a_j}_{pos,i} + D^{a_j}_{neg,i})\lambda^{i-1}] + 2} \quad (8)$$

Finally the overall trustworthiness of the seller $s$ may be calculated:

$$w' = \begin{cases} \frac{N^b_{all}}{N_{min}} & \text{if } N^b_{all} < N_{min} \\ 1 & \text{otherwise} \end{cases}$$

$$Tr(s) = w'R_{pri}(s) + (1 - w')R_{pub}(s) \quad (9)$$

where $N_{min}$, the minimum number of ratings needed for the buyer $b$ to be confident about the private reputation value it has of the seller $s$, is calculated according to equation 3, but is not necessarily the same value used in equation 4.

The model also includes an incentive mechanism, whereby honest advisors are rewarded by better offers from sellers, and in turn these sellers receive better reputations and ultimately more customers. While interesting in its own right, this mechanism does not directly affect our current work, and therefore we do not discuss this part of Zhang's model further.

### 2.2 Towards a Method for Determining Advisors

The Beta Reputation System (BRS) [3] uses beta population density functions in order to combine feedback from multiple sources and subsequently derive reputation ratings. The initial evaluation gives some broad estimates about the number of ratings required to obtain an accurate group rating; however, these are at best snapshots of what size network *might* be needed to obtain a stable rating under certain conditions. Ultimately, the evaluation provides little insight into what an optimal network size would be under the BRS.

TRAVOS [4] calculates trust using probability theory, taking into account any past interactions between agents, or alternatively reputation information provided by third parties. Here, each agent is expected to maintain a trust assessment for every other agent in the system that it has interacted with in the past. Under the circumstances, it seems that restricting the size of the advisor network is unrealistic. Indeed, to our knowledge, nothing of this nature have been explored with respect to the TRAVOS system.

Some potential methods for limiting the number of advisors may be derived from the work done in [2] to evaluate various design choices in a collaborative filtering algorithm. The first method, *correlation thresholding* [5], sets an minimum correlation weight that an advisor must have in order to be considered part of the user's "neighbourhood". However, if the threshold is set too high, then the neighbourhood may be very small, limiting the possibilities for predictions. In fact, for the data set examined in [2], correlation thresholding yielded declines in both coverage and accuracy compared to a non-thresholded algorithm.

The second method discussed, *best-n-neighbors*, as used in the GroupLens [6] system among others, picks a maximum number of neighbours to use, *max_nbors*. The neighbours chosen would be those with the highest correlation to the instant user. In [2], it was shown that a neighbourhood of 20 to 50 users (out of a population of 943) was found to provide an acceptable level of accuracy, providing an appropriate balance between sufficient coverage and eliminating inaccuracies.

A potential supplement to finding the optimal number of advisors is derived from [7], which discussed reputation management in a social network making use of an *advisor referral* mechanism. In this mechanism, a buyer's agent would consult its "neighbour" agents, each of which might either provide advice on the question itself, provide references to other appropriate advisors, or both, depending on the question. As a result, a buyer would be able to benefit from the information held by the pool of advisors without having a large number of neighbours [1]. It then stands to reason that this method could be used in combination with network-size optimization to provide a smaller advisor network size.

## 3 Analysis

### 3.1 Limiting the Network Size

The results in [2] would appear to point towards setting a maximum number of advisors as the preferable method of restricting the advisor network size, as opposed to correlation (or, in our case, trust) thresholding. However, we cannot overlook the distinction between correlation for collaborative filtering and reputation. While similarity with a buyer may indirectly impact on that buyer's private reputation of an advisor, the private reputation of a seller only relates to the buyer's ratings for that seller, ignoring similarity, while similarity is not a factor at all in public reputation. Hence we propose that both options, trustworthiness thresholding and maximum number of advisors, should be thoroughly examined.

That said, neither technique could be directly applied to Zhang's model for advisor reputation; the trustworthiness values must be calculated for all possible advisors before the buyer can proceed to calculate seller reputation.

Our application of these techniques in the seller reputation model are formalized as follows.

**Trustworthiness Thresholding** Choose some threshold $L$ $(0 \leq L \leq 1)$ which represents the minimum advisor trustworthiness value $Tr(a)$ required to be included in the advisor network. We then define the set $A_{L,b} = \{a_1, a_2, \ldots, a_k\}$ consisting of all advisors for which $Tr(a) \geq L$ for a particular buyer $b$. We then use the subset $A_{L,b,s}$, consisting of the advisors in $A_{L,b}$ that have provided ratings for the seller $s$, in place of the previously-defined set $\{a_1, \ldots, a_k\}$, the set of all advisors that have provided ratings for $s$, in Zhang's Algorithm 2 (the seller reputation algorithm outlined in Equations 7 to 9 in this paper).

**Maximum Number of Advisors** For a particular buyer $b$, after having calculated the personalized trustworthiness of each advisor for $b$ as per the first part of Zhang's model, we sort the list of all $n$ advisors from greatest trustworthiness value to least, in the set $\{a_1, a_2, \ldots, a_n\}$. We choose some maximum number of advisors for each buyer, $max\_nbors \leq n$, and then truncate this set to the set $A_b = \{a_1, a_2, \ldots, a_{max\_nbors}\}$. We thus obtain the set of $max\_nbors$ advisors that have been calculated to be the most trustworthy for $b$. Again, the subset of $A_b$ that has provided ratings for the seller $s$ is used in place of the larger set $\{a_1, \ldots, a_k\}$ in Zhang's Algorithm 2.

### 3.2 Advisor Referrals

We also wish to consider the possibility of combining one or both of the above methods with the advisor-referral technique suggested in [7] and discussed above. We diverge somewhat from the original mechanism insofar as Zhang's model does not require us to query each advisor for a recommendation. Rather, the buyer has access to each advisor's ratings for a given seller $s$ via a central server, and uses this data to determine the public (or network) reputation for the seller.

We thus consider that advisors can "advise" by allowing buyers to make use of each advisor's own private reputation for a certain seller. In this case, an advisor "referral" system could be implemented using a variant of the measure used to weight private reputation in the original model. This would work as follows: For each advisor $a_j$ in the advisor network of $b$, that is, the set $\{a_1, a_2, \ldots, a_k\}$, $b$ checks whether advisor $a_j$ is an acceptable advisor for the seller $s$. This will be the case if $N_{all}^{a_j} \geq N_{min}$, where $N_{all}^{a_j}$ is the number of ratings provided by an advisor $a_j$ for $s$, and $N_{min}$ is some minimum number of ratings (which may be calculated using equation 3).

If $a_j$ is not an acceptable advisor (that is, if $N_{all}^{a_j} < N_{min}$), the algorithm will query $a_j$'s advisor network, sorted from most trustworthy to least trustworthy from the perspective of $a_j$, in order to determine, in a similar fashion, which (if any) of these advisors meet the criteria to be a suitable advisor for $s$. The first such advisor encountered that is itself not either (a) already in the set of acceptable advisors; or (b) in $A_b$ — since this would imply that the recommended advisor would be added in any event at a later stage — will be accepted.

If none of the advisors of $a_j$ meet the above criteria, this step would be repeated at each subsequent level of the network — that is, the advisors of each member of the set of advisors just considered — until an acceptable, unduplicated advisor was identified.

Once the full set of acceptable advisors has been determined, the "network" reputation would be calculated as in Zhang's model, using the advisor trustworthiness values held by the buyer $b$. This, of course, assumes that the seller $s$ has had sufficient past interactions with the various advisors in the network such that there are at least $k$ buyers that have each had at least $N_{min}$ interactions with $s$, which is not guaranteed. If only a smaller number of acceptable advisors can be found, the system will simply use this reduced set to determine the network reputation.

To ensure broad coverage of the network while preventing infinite recursion, we limit the number of network "levels" calculated to at most $\lceil log_k(|B|) \rceil$, where $B$ is the set of all buyers (advisors) in the system. However, we note that practically, in a large scale system, the number of levels may need to be smaller in order for this algorithm to be computationally efficient; we will leave such a decision for later work.

We summarize this mechanism in pseudo-code format as Algorithm 1.

## 4 Examples

### 4.1 Using Zhang's Model

As in [1], we consider the case where a buyer $b$ wishes to assess the trustworthiness of a particular seller $s_0$ with whom the buyer has had little or no experience. For the purposes of this simplified example, we assume that there are three available advisors from which $s_0$ may seek advice, namely $a_x$, $a_y$, and $a_z$.

We assume initially that, among sellers that $b$ has had past dealings with, each of these advisors has provided ratings only for the five sellers $(s_1, s_2, s_3, s_4, s_5)$, and has rated each of the sellers at most once in each time window in the sequence $T$, where $T_1$ is the most recent time window. The ratings may be either positive (1) or negative (0); a dash (-) indicates that no rating was provided during the indicated time window. The ratings provided by each advisor for these sellers are listed in Table 1. The buyer $b$ has also provided some ratings for the sellers, as indicated in the same table; note here that $b$ does not provide ratings for every seller each time window.

**Algorithm 1** Selecting Advisors to Buyer $b$ for Trustworthiness of Seller $s$ Using Referrals

```
$A_b = \{a_1, a_2, \ldots, a_k\}$; {advisors in $b$'s advisor network}
$A_s = \{\}$; {set of advisors that are suitable for providing advice regarding seller $s$}
$N_{min}$ = minimum number of ratings for $a$ to be a suitable advisor regarding $s$;
$maxnetlevel = \lceil log_k(|B|) \rceil$ {the maximum number of search iterations}
for $j = 1$ to $k$ do
  $N_{all}^{a_j}$ = total number of ratings provided by $a_j$ for $s$;
  if $N_{all}^{a_j} \geq N_{min}$ then
    append $a_j$ to $A_s$;
  else
    $netlevel = 2$; {the number of connections between $b$ and the advisors being searched}
    $a_x$ = null; {the desired suitable advisor in place of $a_k$}
    $A_c$ = the set of advisors for $a_j$ sorted from most to least trustworthy (as per $a_j$);
    while $a_x$ == null and $netlevel \leq maxnetlevel$ do
      $A_n = \{\}$; {the set of advisors to be considered in the next round, if necessary}
      for all $a_c$ in $A_c$ do
        $N_{all}^{a_c}$ = total number of ratings provided by $a_c$ for $s$;
        if $N_{all}^{a_c} \geq N_{min}$ and $a_c \notin A_b$ and $a_c \notin A_s$ then
          $a_x = a_c$;
          break;
        else
          add the set of advisors for $a_c$ to $A_n$;
        end if
      end for
      $netlevel + +$;
      $A_c = A_n$
    end while
    if $a_x \neq$ null then
      append $a_x$ to $A_s$;
    end if
  end if
end for
```

**Table 1.** Ratings of Sellers Provided by Advisors and Buyer $b$

| | $a_w$ | | | | | $a_x$ | | | | | $a_y$ | | | | | $a_z$ | | | | | $b$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $T$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ |
| $s_1$ | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| $s_2$ | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | - |
| $s_3$ | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | - | - |
| $s_4$ | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | - | - | - |
| $s_5$ | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | - | - | - | - |

We derive the trustworthiness values for each advisor using equations 1 through 5. For simplicity, in these calculations, we follow the method used in the examples provided in [1]. First, we will only consider pairs of ratings provided during the same time window, and thus assume that the forgetting factor as defined previously is $\lambda = 0$. For the determination of $N_c$, we assume for simplicity that any rating of 1 provided by the advisor is a "consistent" rating. Finally, in equation 3 we use $\gamma = 0.8$ and $\epsilon = 0.15$, leading to $N_{min} = 51$. The pertinent values are shown in Table 2.

**Table 2.** Trustworthiness of Advisors $a_w$, $a_x$, $a_y$, and $a_z$ for Buyer $b$

| $a_j$ | $N_p$ | $N_{all}$ | $\alpha$ | $\beta$ | $R_{pri}$ | $N_c$ | $N'_{all}$ | $\alpha'$ | $\beta'$ | $R_{pub}$ | $w$ | $Tr(a)$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $a_w$ | 5 | 15 | 6 | 11 | 0.353 | 14 | 25 | 15 | 12 | 0.556 | 0.294 | 0.497 |
| $a_x$ | 15 | 15 | 16 | 1 | 0.941 | 25 | 25 | 26 | 1 | 0.963 | 0.294 | 0.957 |
| $a_y$ | 8 | 15 | 19 | 8 | 0.529 | 11 | 25 | 12 | 15 | 0.444 | 0.294 | 0.469 |
| $a_z$ | 0 | 15 | 1 | 16 | 0.059 | 0 | 25 | 1 | 26 | 0.037 | 0.294 | 0.0434 |

We proceed to the calculation the trustworthiness of a seller $s_0$. As a preliminary matter, we remember that the buyer $b$ has not provided any ratings in the past for $s_0$, and therefore $R_{pri}(s) = \frac{1}{2}$. Of our four advisors, only $a_w$, $a_x$ and $a_z$ have provided ratings for the seller $s_0$, as indicated in Table 3(a). The subsequent Table 3(b) indicates how these ratings translate into positive and negative amounts, while Table 3(c) shows how these ratings are discounted based on the advisor trustworthiness values calculated earlier.

**Table 3.** Ratings of $s_0$ Provided by $a_w$, $a_x$, $a_z$

(a) Ratings

| $T_i$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ |
|---|---|---|---|---|---|
| $a_w$ | 1 | 0 | 1 | 0 | 1 |
| $a_x$ | 0 | 0 | 0 | 1 | 1 |
| $a_z$ | 1 | 1 | 1 | 1 | 1 |

(b) Amounts of Ratings

| $T_i$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ |
|---|---|---|---|---|---|
| $N^{a_w}_{pos,i}$ | 1 | 0 | 1 | 0 | 1 |
| $N^{a_w}_{neg,i}$ | 0 | 1 | 0 | 1 | 0 |
| $N^{a_x}_{pos,i}$ | 0 | 0 | 0 | 1 | 1 |
| $N^{a_x}_{neg,i}$ | 1 | 1 | 1 | 0 | 0 |
| $N^{a_z}_{pos,i}$ | 1 | 1 | 1 | 1 | 1 |
| $N^{a_z}_{neg,i}$ | 0 | 0 | 0 | 0 | 0 |

(c) Discounted Amounts of Ratings

| $T_i$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ |
|---|---|---|---|---|---|
| $D^{a_w}_{pos,i}$ | 0.397 | 0 | 0.397 | 0 | 0.397 |
| $D^{a_w}_{neg,i}$ | 0 | 0.397 | 0 | 0.397 | 0 |
| $D^{a_x}_{pos,i}$ | 0 | 0 | 0 | 0.937 | 0.937 |
| $D^{a_x}_{neg,i}$ | 0.937 | 0.937 | 0.937 | 0 | 0 |
| $D^{a_z}_{pos,i}$ | 0.0294 | 0.0294 | 0.0294 | 0.0294 | 0.0294 |
| $D^{a_z}_{neg,i}$ | 0 | 0 | 0 | 0 | 0 |

Using equation 8, we may then find the public reputation of $s_0$. In keeping with the examples provided in [1], we remove our previously-stated simplification that only compared ratings in the same time window, and thus set a forgetting factor of $\lambda = 0.9$:

$$R_{pub}(s_0) = \frac{\sum_{i=4}^{5} 0.937 * 0.9^{i-1} + 0.397 * (0.9^0 + 0.9^2 + 0.9^4) + \sum_{i=1}^{5} 0.0294 * 0.9^{i-1} + 1}{\sum_{i=1}^{5} 0.937 * 0.9^{i-1} + \sum_{i=1}^{5} 0.397 * 0.9^{i-1} \sum_{i=1}^{5} 0.0294 * 0.9^{i-1} + 2} = 0.4480$$

Finally, since the buyer has not dealt with $s_0$ before, the weight for the private reputation $w'$ is zero, meaning we can immediately conclude that $Tr(s_0) = 0.4480$.

### 4.2 Reputation Thresholding

We now turn to exploring the effects of the modifications proposed in this paper by first examining how setting a minimum reputation threshold would affect the size of our network and our seller reputation model. We choose several potential values for the threshold $L$ and indicate, based on the results in the previous section regarding advisor trustworthiness, how many advisors would be included in the buyer $b$'s advisor network in this case. The results are shown in Table 4.

**Table 4.** Advisor Network Size with a Correlation Threshold

| $L$ | 0 | 0.2 | 0.4 | 0.6 | 0.8 | 1 |
|---|---|---|---|---|---|---|
| $A_{L,b}$ | $\{a_w, a_x, a_y, a_z\}$ | $\{a_w, a_x, a_y\}$ | $\{a_w, a_x, a_y\}$ | $\{a_x\}$ | $\{a_x\}$ | $\{\}$ |
| $\lvert A_{L,b} \rvert$ | 4 | 3 | 3 | 1 | 1 | 0 |

Trivially, when $L = 0$, all advisors will be included in the network, and $Tr(s_0) = 0.4480$. For $L = 0.2$ and $L = 0.4$, the advisor network consists of $a_w$, $a_x$, and $a_y$, of which only $a_w$ and $a_x$ contribute ratings for $s_0$. We refer to the resulting trustworthiness value as $Tr_{w,x}(s_0)$, which we calculate using Equation 8 to be 0.439.

For $L = 0.6$ and $L = 0.8$, the advisor network consists solely of $a_x$, and therefore the seller trustworthiness $Tr_x(s_0)$ (again by Equation 8) would be 0.697. Finally, for $L = 1$, the advisor network is the empty set and, trivially, $Tr_{empty}(s_0) = \frac{1}{2}$.

### 4.3 Maximum Number of Advisors

If we instead elect to use a maximum number of advisors, we would have the results shown in table 5, with the advisor network representing the $max\_nbors$ advisors most trusted by the buyer $b$. For comparison, we indicate the minimum trustworthiness value of the advisors in the network, to show the maximum threshold $L$ that could be used to get the same result using thresholding.

**Table 5.** Trustworthiness of $s_0$ Using a Maximum Number of Advisors

| $max\_nbors$ | 0 | 1 | 2 | 3 | $\geq 4$ |
|---|---|---|---|---|---|
| $A_b$ | $\{\}$ | $\{a_x\}$ | $\{a_w, a_x\}$ | $\{a_w, a_x, a_y\}$ | $\{a_w, a_x, a_y, a_z\}$ |
| $min(Tr(a))$ | undefined | 0.957 | 0.497 | 0.469 | 0.0434 |
| $Tr(s_0)$ | 0.5 | 0.697 | 0.439 | 0.439 | 0.4480 |

### 4.4 Advisor Referrals

We now examine the addition of our advisor referral mechanism to this system. To do so, we introduce a new advisor into the system, $a_v$, as well as another seller, $s_6$. To this point $a_v$ has only provided ratings for $s_6$, while $b$ has not provided any ratings for that seller; therefore there are no commonly-rated sellers for $a_v$ and $b$, and thus $Tr(a_v) = 0.5$ from the perspective of $b$.

We also assume, as in the $max\_nbors = 3$ or $L = 0.4$ cases described above, that the advisor network for $b$ consists of $\{a_w, a_x, a_y\}$ — $a_v$ is too new to have been considered as a potential advisor in that case, although for purposes of demonstration we assume that $a_v$ has somehow been included in the advisor networks of some of the other advisors. Finally we set $N_{min}$, the minimum number of ratings for an advisor to be considered acceptable for a given seller, as 3. The ratings that have been given by each advisor, and the resulting discounted amounts, are as shown in Table 6.

Given this information, the buyer $b$ will examine its advisor network and find that $a_w$ and $a_x$ are indeed acceptable advisors for $s_6$, since both have achieved at least $N_{min}$ interactions with $s_6$. However, $a_y$ has only had one interaction with $s_6$, and would therefore not be considered an acceptable advisor. The buyer will then look to $a_y$'s advisor network to identify an appropriate substitute.

Suppose then that $a_y$ also has a three-agent advisor network consisting of $a_v$, $a_x$, and $a_z$, with trustworthiness values 0.5, 0.6, and 0.7 respectively. This information will be gathered by $b$ as the ordered list $\{a_z, a_x, a_v\}$. The buyer will then iterate through the set, discarding

**Table 6.** Ratings of $s_6$ Provided by Advisors

(a) Ratings

| $T_i$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ |
|---|---|---|---|---|---|
| $a_v$ | 1 | 1 | 0 | 1 | 1 |
| $a_w$ | 0 | 1 | 1 | 0 | - |
| $a_x$ | 1 | 0 | 1 | - | - |
| $a_y$ | 0 | - | - | - | - |
| $a_z$ | 1 | 1 | - | - | - |

(b) Amounts of Ratings

| $T_i$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ |
|---|---|---|---|---|---|
| $N^{a_v}_{pos,i}$ | 1 | 1 | 0 | 1 | 1 |
| $N^{a_v}_{neg,i}$ | 0 | 0 | 1 | 0 | 0 |
| $N^{a_w}_{pos,i}$ | 0 | 1 | 1 | 0 | - |
| $N^{a_w}_{neg,i}$ | 1 | 0 | 0 | 1 | - |
| $N^{a_x}_{pos,i}$ | 1 | 0 | 1 | - | - |
| $N^{a_x}_{neg,i}$ | 0 | 1 | 0 | - | - |
| $N^{a_y}_{pos,i}$ | 0 | - | - | - | - |
| $N^{a_y}_{neg,i}$ | 1 | - | - | - | - |
| $N^{a_z}_{pos,i}$ | 1 | 1 | - | - | - |
| $N^{a_z}_{neg,i}$ | 0 | 0 | - | - | - |

(c) Discounted Amounts of Ratings

| $T_i$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ |
|---|---|---|---|---|---|
| $D^{a_v}_{pos,i}$ | 0.4 | 0.4 | 0 | 0.4 | 0.4 |
| $D^{a_v}_{neg,i}$ | 0 | 0 | 0.4 | 0 | 0 |
| $D^{a_w}_{pos,i}$ | 0 | 0.397 | 0.397 | 0 | - |
| $D^{a_w}_{neg,i}$ | 0.397 | 0 | 0 | 0.397 | - |
| $D^{a_x}_{pos,i}$ | 0.937 | 0 | 0.937 | - | - |
| $D^{a_x}_{neg,i}$ | 0 | 0.937 | 0 | - | - |
| $D^{a_y}_{pos,i}$ | 0 | - | - | - | - |
| $D^{a_y}_{neg,i}$ | 0.375 | - | - | - | - |
| $D^{a_z}_{pos,i}$ | 0.0294 | 0.0294 | - | - | - |
| $D^{a_z}_{neg,i}$ | 0 | 0 | - | - | - |

$a_z$ as an unacceptable advisor (having provided only two ratings for $s_6$), and also $a_x$ as it is already in $b$'s advisor network. Finally, $b$ would then accept $a_v$ as the third advisor, as it has an acceptable level of experience with $s_6$ but is not part of $b$'s own advisor network.

As in the previous examples, $b$ does not itself have enough experience with $s_6$ to generate a private reputation. Therefore, using the above information for the set of advisors $\{a_v, a_w, a_x\}$, the forgetting factor $\lambda = 0.9$, and Equation 8, we find that $Tr(s_6) = 0.6655$.

If $b$ had not used advisor referrals but instead relied solely on its existing advisor network, namely $\{a_w, a_x, a_y\}$, it would have obtained a significantly different result — $Tr(s_6) = 0.5549$. However, the latter result makes much less use of the experience within the network for $s_6$ than did the one incorporating advisor referrals.

## 5 Discussion

In this paper, we have outlined three potential improvements to Zhang's personalized trust-modelling approach — trustworthiness thresholding, maximum number of advisors, and advisor referrals — all of which aim to reduce the computational complexity required to derive recommendations for trustworthy sellers from its advisors, and to improve the accuracy of these recommendations.

Again, much work remains to show that any or all of these approaches would be effective in improving on Zhang's approach. A more in-depth experiment using a larger set of data is required in order to verify which of these methods will provide the best performance - if, indeed, any of them is superior to the results in Zhang's original model. These will likely be in similar format to the comparative evaluations provided in [1], including marketplace simulations involving a number of buyer / seller agents with varying levels of honesty.

We will first seek to determine optimal parameters, or ranges of parameters, for both the threshold and *max_nbors* methods. Subsequently we will determine how these optimal versions compare to each other and to Zhang's original model. Finally, presuming that at least one of these modifications provides improved performance, we will attempt to implement advisor referrals on that system; again, we may need to try a number of parameter values in order to determine an optimal method.

For each stage of the evaluation, we should consider how well each method performs at correctly distinguishing between honest and dishonest agents. As in [1], we may compare performance using the Matthews correlation coefficient [8], which provides a single measure for the quality of binary classifications, such as for honest and dishonest sellers.

Other open questions remain: we might consider, for example, Zhang's suggestion to apply this model to time-sensitive tasks which may require a buyer to make a very quick decision; here, the buyer would only have time to consult a limited number of advisors.

Finally, we have found very limited work in the past on the effects of the size of the advisor network, or indeed other characteristics such as advisor referrals, on the usage of trust-based approaches of this nature. This research, into improvements to a particular system, may only have limited application to trust in general; it is likely that further work will be required to generate some more concrete principles in this area.

## 6 Acknowledgements

This research was supported in part by funding from the Natural Sciences and Engineering Research Council of Canada.

## References

1. Zhang, J.: Promoting Honesty in E-Marketplaces: Combining Trust Modeling and Incentive Mechanism Design. PhD thesis, University of Waterloo (2009)
2. Herlocker, J., Konstan, J.A., Riedl, J.: An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. Information Retrieval **5**(4) (October 2002) 1386–4564
3. Jøsang, A., Ismail, R.: The beta reputation system. In: 15th Bled Electronic Commerce Conference. (2002)
4. Teacy, W.T.L., Patel, J., Jennings, N.R., Luck, M.: Coping with inaccurate reputation sources: Experimental analysis of a probabilistic trust model. In: Fourth International Joint Conference on Autonomous Agents and Multiagent Systems. (2005) 997–1004
5. Shardanand, U., Maes, P.: Social information filtering: Algorithms for automating "word of mouth". In: Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems. (1995) 210–217
6. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: An open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 Conference on Computer Supported Collaborative Work. (1994)
7. Yu, B., Singh, M.P.: A social mechanism of reputation management in electronic communities. In: Proceedings of Fourth International Workshop on Cooperative Information Agents. (2000) 154–165
8. Matthews, B.W.: Comparison of the predicted and observed secondary structure of t4 phage lysozyme. Biochimica et Biophysica Acta **405**(2) (October 1975) 442–451

# Trust-Based Recommendation Based on Graph Similarity

Chung-Wei Hang and Munindar P. Singh

Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206, USA
{chang,singh}@ncsu.edu

**Abstract.** Trust networks are directed weighted graphs whose nodes represent agents and edges represent trust between agents. This paper proposes a trust-based recommendation approach, which can recommend trustworthy agents to a requester in a trust network. We consider a good recommendation as one to an agent that the requester's trusted neighbors trust highly. We relate the recommendation problem to the graph similarity problem, and define the similarity measurement as a mutually reinforcing relation. By calculating the vertex similarity between the trust network and a structure graph (a path graph of length three), we can produce a recommendation based on similarity scores that reflect both the link structure and the trust values on the edges.

**Key words:** Trust, Graph Similarity, Recommendation

## 1 Introduction

Trust networks are directed weighted graphs whose nodes represent agents, edges represent trust relations, and weights represent trust values [1, 2]. An edge from node $u$ to node $v$ with trust value $t$ means $u$ trusts $v$ to the extent of $t$. A similar concept is commonly seen in the real world. Examples include Facebook[1] where edges are the friendships between people; citation networks where nodes are papers and edges are citations; web graphs where nodes are webpages and edges are hyperlinks; FilmTrust [3] where edges are movie taste similarity between people; and, Epinions [4] and Advogato [5, 6], where the edges are trust relations.

There are two main challenges in trust networks: (a) *trust propagation*, and (b) *trust-based recommendation*. Trust propagation is about predicting the trustworthiness of nonadjacent agents by combining trust values through distinct indirect paths. To be more specific, trust propagation defines how trust values are aggregated and propagated through a trust network. It can help agents estimate a stranger's trustworthiness without assuming previous experience with the stranger. The other problem in trust networks is trust-based recommendation. Given a trust network and a target agent $v$, how to recommend a trustworthy

[1] http://www.facebook.com/

agent for $v$ to interact with. Trust propagation is widely studied in the literature [1, 5, 7, 8, 6, 2, 9, 10], whereas trust-based recommendation has not drawn much attention from the research community.

A possible solution to trust-based recommendation is to apply trust propagation to estimate the trustworthiness of all agents that are not adjacent to the target $v$, and recommend the agents with high trust estimates. However, this solution is not promising because the complexity of the trust propagation grows quickly as the number of agents increases. Another possible solution is to recommend agents who share a fair number of common neighbors. For example, Facebook recommends friends based on the number of mutual friends between people. However, this approach fails to take the trust values (i.e., edge weights) into consideration.

Our approach aims to provide a trust-based recommendation approach, which recommends trustworthy relationships by considering not only the link structure (e.g., the number of common neighbors), but also the trust values associated with the links. Instead of considering one single potential neighbor separately, our trust-based recommendation processes the trust network around the target (i.e., the agent that requests a recommendation), thereby providing recommendations more efficiently. The idea behind our approach is based on graph similarity [11]. We show that by calculating vertex similarity between the trust network and a *structure graph*, the trust-based recommendation problem can be translated into a graph similarity problem, and the similarity scores can be viewed as a measurement of how many good connections (i.e., with high trust values) the agents shares with the target. Besides, instead of predicting how the trust network evolves from a *network*-level perspective, our approach departs from a *node*-level perspective, providing personalized recommendations, especially for the target.

Note that we can customize trust-based recommendation by using different structure graphs. In this paper, we study two basic structure graphs that facilitate making recommendation based on in-degree and friends of friends, respectively. However, our approach is not limited to these structure graphs. It can be extended to produce recommendations based on other criteria.

To summarize, our trust-based recommendation takes a trust network and a target agent (who requests recommendation in the trust network) as inputs, and outputs a list of trustworthy agents for the target to interact with as recommendation. This paper makes four key contributions. First, our approach provides personalized recommendations for a particular agent. Second, a recommendation produced is based not only on the network topology, but also on the trust values associated with the edges. Third, our approach is efficient because it considers only a small subgraph of the trust network, and processes potential candidates all at once. Fourth, our approach allows for customizing recommendation based on various criteria.

The rest of paper is organized as follows. Section 2 surveys the state of the art of the related research areas. Section 3 presents our approach by first introducing the graph similarity measurement used in our approach, and then demonstrating how the trust-based recommendation problem can be solved via graph similarity. Section 4 concludes this paper and identifies possible future directions.

## 2 Related Work

Here we categorize the literature into four areas: *graph similarity*, *link prediction*, *trust propagation*, and *recommender systems*.

Our trust-based recommendation approach is built on a graph similarity measurement. Graph similarity has been applied in various applications. For example, Melnik et al. [12] present a graph similarity approach, called *similarity flooding*, for database schema matching. Their approach takes two graphs as inputs, measures the vertex similarity between the inputs, and outputs a mapping—a subgraph consisting of similar nodes. This work differs from ours in many ways except both apply vertex similarity between two graphs. First, Melnik et al.'s input graphs have no weights, whereas our approach takes the edge weights into consideration. Second, their approach takes two graphs as input, and calculates the similarity between them. Our approach takes only one graph as input. Given that graph, our approach calculates the similarity between the graph and a *structure graph*, which reflects the features we care about in the recommendation. Third, Melnik et al.'s approach requires adjustment by humans, which ours does not.

Jeh and Widom [13] propose a domain-independent similarity measurement, *SimRank*. SimRank measures the similarity between objects. It follows the intuition that "two objects are similar if they are related to similar objects." Jeh and Widom first convert the graph to a *node-pair* graph, where each node represents a node-pair in the original graph. The node-pair $(a, b)$ is connected to the node-pair $(c, d)$ if $a$ connects to $c$ and $b$ connects to $d$ in the original graph. Then they calculate and propagate similarity score in the converted graph iteratively until convergence. Again, Jeh and Widom only consider graphs with no edge weights, whereas edge weights (i.e., trust values) play an important role in our approach.

Link prediction for large networks studies how to predict the edges that will be added in the future, given the current snapshot of a network. Liben-Nowell and Kleinberg [14] survey various link prediction methods from graph theory and social-network analysis. These methods measure the similarity between nodes with respect to the network topology, assign a weight to each pair of nodes, and generate a list sorted in decreasing order in terms of weights. Liben-Nowell and Kleinberg evaluate these link prediction methods in five collaboration networks where edges connect authors who coauthor papers. They indicate that the link prediction approaches can provide a network evolution model learned from the observed data. This learned network evolution model can infer how the network is going to evolve based on the network features. Unfortunately, Liben-Nowell and Kleinberg's approach only considers undirected graphs without edge weights.

Kunegis and Lommatzsch [15] propose a general link prediction approach, which applies machine learning techniques to reduce the learning parameters for link prediction, and then uses a curve-fitting method to estimate the parameters. Their approach can be applied to undirected, weighted, or bipartite graphs. Kunegis and Lommatzsch evaluate the approach on web graphs, trust networks, social networks, citation networks, and collaboration networks. Note that, in general, the link prediction methods provide *network*-level prediction, whereas our approach focuses on *node*-level recommendation. In other words, link prediction recommends links for the whole network, but our approach recommends trustworthy others for a particular node.

Trust propagation provides an alternative solution to trust-based recommendation from a node-level perspective. Recommendations can be made by first estimating the trustworthiness of all nonadjacent agents, and then listing the agents with high trust estimates. Hang et al. [10] model trust as a binary event. They define three operators for concatenating trust along a path, aggregating trust from distinct paths from the same witness, and selecting the most trustworthy path among all witnesses, respectively. Advogato [5] adopts a network flow algorithm where the flow capacity of edges is determined by the depth along the path. Appleseed [6] applies spreading activation, where *trust energy* is spread across the trust network. The energy is divided when the agent has more than one successor. All these trust propagation methods provide recommendation for a particular agent. However, they are not computationally efficient because the complexity grows quickly as the number of agents increases. Our approach does not treat each of the nonadjacent agents separately. Instead, it processes all agents at the same time, yielding better performance.

Now we discuss some related work of recommender systems. In general, recommender systems suggest *items* to *users*. There are two main categories of recommender systems: *content-based* and *collaborative filtering* systems [16]. Content-based approaches produce recommendations based on the similarity between items. Collaborative filtering approaches recommend the items chosen by the users with similar tastes. Our approach is closer to a collaborative filtering approach because some of the collaborative filtering approaches construct a trust network where nodes are users and edges represent the similarity between users' taste. For example, FilmTrust [3] is a social network where edges represent the similarity of movie taste. Ben-Shimon et al. [17] propose a recommendation approach that is quite similar to ours. They construct a *personal* social network containing friends of friends (up to six levels) of a user who needs recommendation. Ben-Shimon et al. then find the sum of all the ratings of a particular item, discounted by the distance from the rater to the user. If the sum is high, the item is recommended. There are no items involved in our case. Thus, rather than computing the sum of all ratings, our approach considers the link structure and the trust values on the edges. Fouss et al. [18] present a recommender system based on a similarity measurement between the nodes of a directed weighted graph. They compute similarity based on a Markov-chain random walk model, which assigns a transition probability to each edge. The distance required for a random walker to travel from one node to another defines the similarity between these two nodes. Our approach is different from Fouss et al.'s approach in two ways. First, our approach considers directed graphs rather than undirected ones. Second, we use the similarity measurement defined by a *mutually reinforcing relation* rather than the Markov-chain random walk model.

## 3 Approach

Now we introduce our approach. In Section 3.1, we briefly overview the graph similarity measurement applied in our approach. Section 3.1 also shows two applications of the graph similarity by customizing different structure graphs. In Section 3.2, we first define a trust network. Next we customize the graph similarity measurement by devising a structure graph that satisfies our claim for suggesting recommendations in a trust network. Then we show how trust values are considered, and how they affect the recommendations produced. We formalize our approach at the end.

### 3.1 Background: Vertex Similarity between Graphs

Blondel et al. [11] propose a vertex similarity measurement between graphs. Given two directed graph $G_A$ with $n_A$ vertices, and $G_B$ with $n_B$ vertices, a similarity matrix $\mathbf{S}$ is a $n_A \times n_B$ matrix where $s_{ij}$ is the similarity score between node $i$ in $G_A$ and node $j$ in $G_B$. $\mathbf{S}$ can be calculated by a convergent iterative process:

$$\mathbf{S}_{k+1} = \frac{B\mathbf{S}_k A^T + B^T \mathbf{S}_k A}{\|B\mathbf{S}_k A^T + B^T \mathbf{S}_k A\|_F}, \tag{1}$$

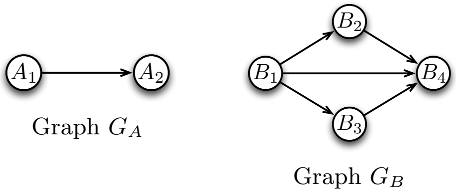
where $A$ and $B$ are the adjacency matrices of $G_A$ and $G_B$, respectively, $\mathbf{S}_0$ has all entries equal to 1, and $\|.\|_F$ is the square root of the sum of the squares of all entries. The denominator normalizes $\mathbf{S}_{k+1}$ to $[0, 1]$. The limit of this convergent process is $\mathbf{S}$. The convergence can be determined by

$$\|\mathbf{S}_{k+1} - \mathbf{S}_k\|_F < \epsilon, \tag{2}$$

where $\epsilon$ is the error tolerance.

For example, Figure 1 shows the similarity matrix between two graphs: $G_A$ and $G_B$. $G_A$ contains two vertices: $A_1$ has out-degree of one, and $A_2$ has in-degree of one. After measuring the vertex similarity with $G_B$, one can observe that $B_1$, which has the largest out-degree, is the most similar vertex to $A_1$. $B_4$, which has the largest in-degree, has the highest similarity score to $A_2$. Notice that the greater the out-degree a vertex has, the higher its similarity score to $A_1$. An analogous observation applies to the in-degree and $A_2$. We can conclude that by comparing the similarity score to $G_A$, we can find the vertex that connects to the most others, and is connected by the most others.

The idea behind the similarity measurement is the *mutually reinforcing relation*, which is widely applied in web search [19, 20], and reputation management in peer-to-peer systems [21]. To illustrate the mutually reinforcing relation, we take $G_A$ and $G_B$ in Figure 1 as an example. For each vertex $B_i$ in $G_B$, we associate two similarity scores, say $s_{i1}$ (for $A_1$) and $s_{i2}$ (for $A_2$), each of which



| | $A_1$ | $A_2$ |
|---|---|---|
| $B_1$ | 0.5774 | 0 |
| $B_2$ | 0.2887 | 0.2887 |
| $B_3$ | 0.2887 | 0.2887 |
| $B_4$ | 0 | 0.5774 |

**Fig. 1.** Example of two graphs, $G_A$ and $G_B$, and their similarity matrix. The most similar vertex to $A_1$ is $B_1$, which has the largest out-degree; the most similar vertex to $A_2$ is $B_4$, which has the largest in-degree.

corresponds to the similarity between one vertex in $G_A$ and $B_i$. Both scores are initialized to one. Then the scores are updated according to the mutually reinforcing relation:

$$\begin{cases} s_{i1} = \sum_{j:(i,j)\in E_B} s_{j2} \\ s_{i2} = \sum_{j:(j,i)\in E_B} s_{j1} \end{cases} \tag{3}$$

This mutually reinforcing relation says a vertex is similar to $A_1$ if it connects to many vertices that are similar to $A_2$, whereas a vertex is similar to $A_2$ if it is connected by many vertices that are similar to $A_1$. The update process is iterated. The scores $s_{i1}$ and $s_{i2}$ mutually reinforce each other. Blondel et al. show that this update process converges to a state, which corresponds to the similarity scores between $A_1$ and $A_2$, and $B_i$.

Now let us extend the similarity scores to all vertices in $G_B$. Suppose $\mathbf{s}_1$ and $\mathbf{s}_2$ are the similarity scores to $A_1$ and $A_2$, respectively, for all $B_i$ in $G_B$

$$\mathbf{S}_{k+1} = \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix}_{k+1} = \begin{bmatrix} 0 & B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix}_k = \mathbf{S}_k, \tag{4}$$

where $B$ is the adjacency matrix of $G_B$ and $k = 0, 1, \ldots$. Blondel et al. further extend $G_A$ to an arbitrary graph and simplify Equation 4 to Equation 1, where $A$ is the adjacency matrix of $G_A$.

By using different $G_A$ (called the *structure graph*), we can apply the vertex similarity to solve different problems and applications. Blondel et al. show that the web search algorithm, *HITS* [20], which searches for webpages based on a query, is an application of vertex similarity. HITS ranks webpages based on an *authority score* and a *hit score*. A webpage is a good authority if there are many hits that link to it. In contrast, a good hit is a webpage that points to many good authorities. The HITS algorithm is a special case that compares the vertex similarity between the cyberspace and $G_A$ in Figure 1. Blondel et al. point out another application, synonym extraction. They construct a directed graph from a dictionary, where a node represents a word, and an edge from $u$ to $v$ means $u$ is used in the definition of $v$. They first create a subgraph of the dictionary graph by extracting all words connecting to a word $x$ or connected by $x$. Then they calculate the similarity score with the graph $G_s$ shown in Figure 2. The vertices with higher similarity score to $A_2$ are chosen as synonyms of $x$. The similarity score to $A_2$ of a word $y$ indicates how many common words occur in $x$ and $y$'s definitions, and how many definitions use both $x$ and $y$. Thus, the similarity score to $A_2$ reflects the definition of a synonym.

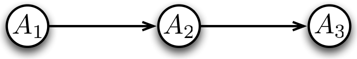


**Fig. 2.** Structure graph $G_S$.

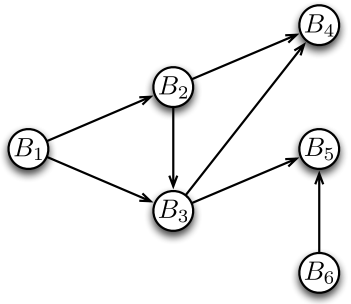
### 3.2 Trust-based Recommendation Based on Graph Similarity

A trust network is a graph where nodes represent agents and edges represent trust relations [1, 2]. A trust relation from agent $u$ to agent $v$ indicates how much trust $u$ places in $v$. Thus, an edge in a trust network is associated with a trust value as its weight. Depending on the trust models, a trust value can be a single scalar, a Beta distribution, or follow another representation. The trust relations can be obtained from a direct interaction or from a referral via trust propagation [10]. For example, a social network such as Facebook is a trust network where all edges are modeled have the same trust values. Here we only consider a trust value as a single scalar. Other trust representations can be translated into a scalar, for example, a probability.

**Definition 1.** *A trust network $TN$ is a directed weighted graph $TN(V, E)$, where $V$ is a finite set of agents $\{v_1, \ldots, v_n\}$, and $E$ is a set of trust relations $\{e_1, \ldots, e_m\}$.*

Consider the recommendation problem in trust networks: given a snapshot of a trust network, how can we recommend (i.e., predict) a trustworthy agent to an agent $v$? By intuition, we claim a good recommendation for $v$ is an agent connected by many of $v$'s neighbors. Let us start with a simple case where all edges have the same trust values 1 (i.e., no weights). For example, Figure 3 (left) shows a trust network $TN_B$, which contains the neighbors of the neighbors of agent $B_1$. Among all the agents except $B_1$'s neighbors $B_2$ and $B_3$, $B_4$ is the most possible candidate, because it is connected by two neighbors of $B_1$. Consider the structure graph $G_S$ in Figure 2, which illustrates our claim of producing good recommendations: a friend ($A_3$) of $A_1$'s friend ($A_2$) is probably $A_1$'s friend (i.e., a good recommendation for $v$ is an agent connected by many of $v$'s neighbors). Figure 3 (right) shows the similarity matrix between $G_S$ and $TN_B$. The similarity score between $A_3$ and vertices indicates how the link structure of the vertices is similar to the link structure of $A_3$.

Now we consider the general case where each of the edges in $TN_B$ is associated with a trust value. Instead of using the adjacency matrix, we define the *adjacency matrix with trust*, which is similar to the adjacency matrix for *multigraphs* (permitted to have multiple edges between the same end nodes), except the entries can be non-integers. The entries in the adjacency matrix with trust



| | $A_1$ | $A_2$ | $A_3$ |
|---|---|---|---|
| $B_1$ | 0.4258 | 0.0901 | 0 |
| $B_2$ | 0.3107 | 0.3562 | 0.0458 |
| $B_3$ | 0.0828 | 0.4809 | 0.2270 |
| $B_4$ | 0 | 0.1300 | 0.4258 |
| $B_5$ | 0 | 0.0329 | 0.2940 |
| $B_6$ | 0.0167 | 0.0971 | 0 |

**Fig. 3.** Example of a trust network with no edge weights, and its similarity matrix with the structure graph $G_S$ in Figure 2. Among friends of $B_1$'s friends (i.e., $B_4$, $B_5$, and $B_6$), $B_4$ is the best recommendation with the highest similarity score with $A_3$.

represent the trust values associated to the corresponding edges. One can regard a trust relation from $u$ to $v$ with a high trust value as there exist many edges from $u$ to $v$.

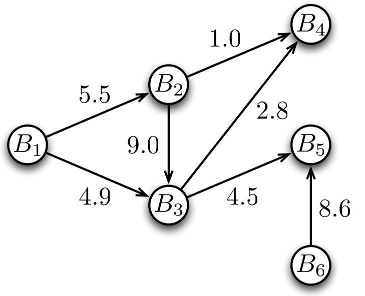
**Definition 2.** *An adjacency matrix with trust $A$ of a trust network $TN(V, E)$ of $n$ agents is an $n \times n$ matrix where the entry $a_{ij}$ is the trust value $v_i$ places in $v_j$.*

Figure 4 shows an example of a trust network $TN'_B$, and the similarity matrix between the structure graph $G_S$ in Figure 2 and $TN'_B$. $TN'_B$ shares the same topology as $TN_B$ in Figure 3, except $TN'_B$ has trust values as its edge weights (rather than 1). Unlike the result in Figure 3, although $B_5$ has fewer connections with $B_1$'s neighbors than $B_4$, $B_5$ has the highest similarity score because the trust value of its only connection is much stronger than the trust values of $B_4$'s connections. Note that $B_3$ (not considered as a recommendation because it is already a neighbor of $B_1$) also has a high similarity score because it is connected by $B_2$ ($B_1$'s neighbor) with a high trust value.

We formalize our trust-based recommendation approach. Given a trust network $TN(V, E)$, to find recommendations for agent $v$, we construct a subgraph $TN'(V', E')$ where $V' \subset V$ contains $v$, all the neighbors of $v$, and the neighbors of $v$'s neighbors, and $E' \subset E$ are all trust relations between any two of $v$, $v$'s neighbors, or the neighbors of $v$'s neighbors. Then the similarity matrix between the structure graph $G_S$ (Figure 2) and $TN'$ is calculated. The nodes that are not neighbors of $v$ and have high similarity scores to $A_3$ are recommended. The reason of taking the subgraph is because if the whole $TN$ is considered, the result will not be a recommendation for the agent $v$. Instead, the agents with the high similarity scores are just similar to $A_3$ in $G_S$, i.e., these agents are connected by many other agents that connected by many others.

We can summarize the main steps of our approach as follows:

1. Given an agent $v$ in a trust network $TN(V, E)$ ($v \in V$).



| | $A_1$ | $A_2$ | $A_3$ |
|---|---|---|---|
| $B_1$ | 0.3950 | 0.1307 | 0 |
| $B_2$ | 0.4044 | 0.3444 | 0.0669 |
| $B_3$ | 0.0089 | 0.4802 | 0.3478 |
| $B_4$ | 0 | 0.0271 | 0.1571 |
| $B_5$ | 0 | 0.0045 | 0.3550 |
| $B_6$ | 0.0036 | 0.1926 | 0 |

**Fig. 4.** Example of a trust network where edge weights represent trust values, and its similarity matrix with the structure graph $G_S$ in Figure 2. Unlike the result in Figure 3, although $B_4$ is connected by $B_1$'s both friends, $B_5$ is the best recommendation because of its strong connection with $B_3$.

2. Construct $TN'(V', E')$, where $V' \subset V$ contains (a) $v$, (b) $v$'s neighbors, and (c) the neighbors of $v$'s neighbors; $E' \subset E$ contains all trust relations between $(u', v') \in V'$.
3. Calculate the similarity matrix $\mathbf{S}$ between $G_S$ (Figure 2) and $TN'$ by Equation 1.
4. Recommend the vertices that are not neighbors of $v$ with high similarity scores to $A_3$ in $G_S$.

## 4 Conclusion

In this paper, we present a trust-based recommendation approach, which provides recommendations to a requester in a trust network. The approach is built on a vertex similarity measurement between graphs. The similarity measurement is defined by a mutual reinforcing relation. We show that by calculating the similarity between the trust network and a structure graph (a path graph of length three), the similarity score can be viewed as a indicator that the agent is strongly connected by the strong neighbors of the requester.

To further validate our approach, a possible future direction is to evaluate on real datasets, for example, FilmTrust [3], Epinions [4], and Advogato [5, 6]. There are three tentative experiment settings. First, we can use cross-validation by removing some of the edges of the requester from the trust network, applying the trust-based recommendation, and comparing the recommendation list with the removed edge list ordered by their trust values. Second, we can trace the evolution of a trust network over time. We can compute the recommendation list based on the past snapshot of the network, and then compare it with the current snapshot to see if the trust network does evolve as predicted. Third, we can compare the recommendation list with the agent list ordered by the estimated trust values calculated by trust propagation [10].

Another direction is to study how our approach can be extended to provide different recommendation. For example, Hang et al. [22] design a trust-based service composition model for estimating trustworthiness of the subservices underlying a composition. However, their model fails to provide a mechanism for selecting the subservices—recommending compositions. Based on our approach, we can construct a structure graph that satisfies their scenario.

## Acknowledgment

This work is supported by the U.S. Army Research Office (ARO) under grant W911NF-08-1-0105 managed by NCSU Secure Open Systems Initiative (SOSI).

## References

1. Yu, B., Singh, M.P.: Distributed reputation management for electronic commerce. Computational Intelligence **18**(4) (November 2002) 535–549
2. Wang, Y., Singh, M.P.: Trust representation and aggregation in a distributed agent system. In: Proceedings of the 21st National Conference on Artificial Intelligence (AAAI), Menlo Park, AAAI Press (2006) 1425–1430
3. Kuter, U., Golbeck, J.: Sunny: A new algorithm for trust inference in social networks using probabilistic confidence models. In: Proceedings of the 22st National Conference on Artificial Intelligence (AAAI), Menlo Park, AAAI Press (2007) 1377–1382
4. Guha, R., Kumar, R., Raghavan, P., Tomkins, A.: Propagation of trust and distrust. In: WWW '04: Proceedings of the 13th international conference on World Wide Web, ACM Press (2004) 403–412
5. Levien, R.: Attack Resistant Trust Metrics. PhD thesis, UC Berkeley (2003)
6. Ziegler, C.N., Lausen, G.: Spreading activation models for trust propagation. In: EEE '04: Proceedings of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service, Washington, DC, USA, IEEE Computer Society (2004) 83–97
7. Richardson, M., Agrawal, R., Domingos, P.: Trust management for the semantic Web. In: The Semantic Web: Proceedings of the 2nd International Semantic Web Conference (ISWC). Volume 2870 of LNCS., Springer (2003) 351–368
8. Gray, E., , Seigneur, J.M., Chen, Y., Jensen, C.: Trust propagation in small worlds. Lecture Notes in Computer Science **2692** (2003) 239–254
9. Quercia, D., Hailes, S., Capra, L.: Lightweight distributed trust propagation. In: Seventh IEEE International Conference on Data Mining (ICDM 2007). (2007) 282–291
10. Hang, C.W., Wang, Y., Singh, M.P.: Operators for propagating trust and their evaluation in social networks. In: Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), Columbia, SC, IFAAMAS (2009) 1025–1032
11. Blondel, V.D., Gajardo, A., Heymans, M., Senellart, P., Dooren, P.V.: A measure of similarity between graph vertices: Applications to synonym extraction and web searching. SIAM Review **46**(4) (2004) 647–666
12. Melnik, S., Garcia-Molina, H., Rahm, E.: Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In: Proceedings of the 18th International Conference on Data Engineering, Washington, DC, USA, IEEE Computer Society (2002)

13. Jeh, G., Widom, J.: SimRank: a measure of structural-context similarity. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM Press (2002) 538–543
14. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. Journal of the American Society for Information Science and Technology **58**(7) (May 2007) 1019–1031
15. Kunegis, J., Lommatzsch, A.: Learning spectral graph transformations for link prediction. In: Proceedings of the 26th Annual International Conference on Machine Learning, New York, NY, USA, ACM Press (2009) 561–568
16. Shani, G., Chickering, M., Meek, C.: Mining recommendations from the web. In: Proceedings of the ACM conference on Recommender systems, New York, NY, USA, ACM Press (2008) 35–42
17. Ben-Shimon, D., Tsikinovsky, A., Rokach, L., Meisels, A., Shani, G., Naamani, L.: Recommender system from personal social networks. In: Proceedings of the 5th Atlantic Web Intelligence Conference, Springer Berlin / Heidelberg (2007) 47–55
18. Fouss, F., Pirotte, A., Renders, J.M., Saerens, M.: Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. IEEE Transactions on Knowledge and Data Engineering **19**(3) (2007) 355–369
19. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems **30**(1–7) (1998) 107–117
20. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. Journal of the ACM **46**(5) (1999) 604–632
21. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The EigenTrust algorithm for reputation management in p2p networks. In: WWW '03: Proceedings of the 12th international conference on World Wide Web, ACM Press (2003) 640–651
22. Hang, C.W., Singh, M.P.: Trustworthy service selection and composition. Technical Report 18, North Carolina State University (July 2009)

# An Architectural Approach to Combining Trust and Reputation

Christopher J. Hazard and Munindar P. Singh

North Carolina State University, Raleigh NC 27695-8206, USA

**Abstract.** Though trust and reputation systems have been extensively studied, general architectural commonalities between the two have received little attention. In this paper, we present a life cycle model of reputation and trust systems, along with accompanying measures of how much effect signaling and sanctioning have on a given system. We map reputation attacks within our framework and apply our framework to an online auction model.

## 1 Introduction

Throughout the trust and reputation system literature, two techniques that stem from game theory are commonly applied for designing such systems. Signaling models are those in which agents attempt to assess private attributes about other agents, whereas sanctioning models are those in which agents behave strategically in an attempt to maximize their utility [4].

In real-world environments where agents must decide whether or not to trust one another, clean distinctions between signaling and sanctioning are rare. For example, an agent that allocates its own bandwidth and other resources may have little influence over the amount of resources it has available. Yet, it may be strategic and rational within those constraints. A manufacturer can acquire a good reputation for having tight quality controls, but new management may wish to see larger profit margins and may strategically slowly cut back on the quality controls as long as it remains ahead of its competitors.

Despite the complexity of the real world, few reputation systems are specifically designed to address both sanctioning and signaling. Typically, authors of reputation systems that involve signaling devise a variety of malicious behaviors to test their system against. Examples of the adversary agents include randomized acts of unfavorable behavior [11, 8], building up and spending of reputation [20, 12, 16], Sybil attacks where an agent creates multiple identities [12, 11, 19], and collusion with other agents [11, 19, 20]. Other systems are designed specifically around strategic agents to ensure good behavior, but do not attempt to measure attributes of the agents [10, 7]. A minority of reputation systems, such as that by Smith and DesJardins [18], examine both signaling and sanctioning explicitly.

Our primary contribution is a model that connects trust and reputation systems both architecturally and functionally. We examine the trust and reputation life cycle in an abstract form from which we can systematically determine how much influence signaling and sanctioning have on the particular system. We present a heuristic that indicates

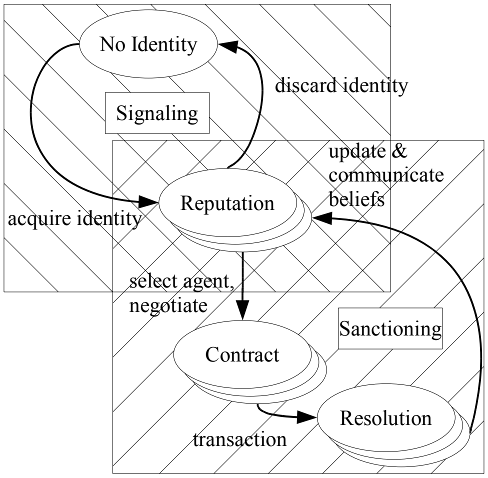


**Fig. 1.** Trust and reputation life cycle from an agent's perspective.

how a system is governed between the signaling and sanctioning, which is prescriptive in terms of what kind of a reputation or trust model should be used for a given situation. From this model, we identify and categorize different kinds of attacks against reputation systems. We use a running example of an online auction.

The general view of this paper is that trust is looking forward in time with respect to sanctioning and strategy, whereas reputation is looking backward in time with respect to signaling and determining agents' types. We discuss this dichotomy in detail. The focus of this work is on rational agents and e-commerce settings, rather than directly modeling human behavior. Emotional and cognitive factors of trust are outside of the scope of this paper.

The remainder of this paper is organized as follows. We first introduce the trust and reputation life cycle using a logic-based formalism and illustrate it via an online auction. We then discuss the signaling and sanctioning dichotomy, measuring the effect of each on a simple online auction model, and then discuss how attacks on reputation systems can occur at each of the points in our model. We conclude with a discussion of the benefits and limitations of our model.

## 2 Trust and Reputation Life Cycle

Although the specifics of particular trust and reputation systems can differ greatly, they all share some commonalities. In this section, we unify the systems to a common set of states and actions as outlined in Figure 1.

### 2.1 Identity States

The following are the different states an agent can go through in a transaction in the presence of an open reputation or trust system. An agent is not limited to being in one state at a time, but can maintain multiple accounts and participate in multiple transactions simultaneously.

**No Identity** The agent begins without an identity or account in the system. This state is applicable for open systems where agents may enter or leave. From this state, an agent may acquire an identity and move to the reputation state. Acquiring an identity may be as trivial as using a nonvalidated screen name in an open text field where the agent simply claims to have some identity. Alternatively, the system may require extensive background checks, verifications from official organizations, or significant payments to create the account. An agent may asynchronously acquire multiple identities, and may acquire identities in different contexts or with different populations of agents.

**Reputation** Each identity that the agent has created will have its own reputation within the community. An agent may discard an identity, either actively by deleting an account or passively by simply no longer using an identity. When an agent decides to (or is forced to) interact with another agent, it must select an agent (or agents) with which to interact. It may communicate with this *target* agent, performing extensive negotiations and setting up a formal contract. Alternatively, the agent may simply rely on norms or not actively communicate with the target prior to the transaction.

**Contract** A contract expresses a promise or commitment to engage in some behavior. Contracts may be well-defined and policed by an external system or may be as ill-defined as the agents' a priori assumptions. From a contract, the agents involved undergo some transaction with the other agents involved. The transaction can involve active participation, such as exchanging money for an item, or a transaction can be passive, such as all agents timing out and not performing any task.

**Resolution** After a transaction has taken place, an agent will update its own beliefs about the agents involved in the interaction. The agent can evaluate, report, and communicate its new beliefs about another agent based on the results of the transaction, either directly to other agents or via a centralized reputation reporting mechanism. Concurrently, the agent may revisit the results and decide that further transactions are required. To set up future transactions, the agents may renegotiate to a new contract after having observed the other agents. A renegotiation can have positive connotations, such as providing additional services to supplement a previous transaction, or the renegotiation can have negative connotations, such as an agent demanding reparations from a transaction that did not fulfill the contract.

### 2.2 Agent Actions

To formalize our discussions about the life cycle of a reputation for further discussion and analysis, we use a logic-based framework. We formally describe abstractions of general interactions of a reputation system where comparisons between values are required to express agents' preferences. For example, we can represent an example of agent Alice's utility as

$$Util(Alice, 5) \wedge Util(Alice, 10) \rightarrow Util(Alice, 15) \tag{1}$$

given the quasilinearity utility rule

$$Util(agent, value1) \wedge Util(agent, value2) \rightarrow Util(agent, value1 + value2). \tag{2}$$

Similar such rules can be used to describe the values within reputation systems. We denote ground terms as those identifiers beginning with an upper case letter, and variables as lower case identifiers.

We can write each of the state transitions from Section 2.1 more formally as follows:

**discard identity** : $ID(agent, id) \wedge Util(agent, discardCost)$
**acquire identity** : $\neg ID(agent, id) \wedge Util(agent, acquireCost)$
**select agent, negotiate** : $Terms(agent, otheragent, contract)$
**transaction** : $Transaction(agent, otheragent, Terms(agent, otheragent, contract))$
**update & communicate beliefs** : $Transaction(agent, otheragent, Terms(agent, otheragent, contract)) \rightarrow Observation(agent, otheragent, terms)$

### 2.3 Example: Online Auction Representation

To illustrate the applicability of our life cycle and formalizations, we create an example Beta model reputation model with only positive and negative ratings resembling an online auction as follows.

Because it costs only a small amount of time for an agent to set up an account, acquiring an identity becomes

$$ID(agent, id) \wedge Util(agent, acquireCost), \tag{3}$$

which could be, for example, $ID(agent, id) \wedge Util(agent, -\$0.5)$. Similarly, discarding an identity is easy, represented as

$$\neg ID(agent, id) \wedge Util(agent, discardCost). \tag{4}$$

The transaction itself becomes

$$\begin{aligned} &Transaction(agent, otheragent, Terms(agent, otheragent, contract)) \rightarrow \\ &\quad \Big(Terms(buyer, seller, contract) \leftrightarrow \big(Pay(buyer, seller, value) \\ &\quad \wedge Give(seller, buyer, good)\big) \vee (CurrentDate > SellDate + 30))\Big), \end{aligned} \tag{5}$$

with

$$Pay(buyer, seller, value) \rightarrow Util(buyer, -value) \wedge Util(seller, value), \tag{6}$$
$$Give(a, b, good) \rightarrow \neg Has(a, good) \wedge Has(b, good), \quad \text{and} \tag{7}$$
$$Has(agent, good) \rightarrow Util(agent, Value(agent, good)). \tag{8}$$

We can represent the ratings mechanism, triggered by the observation of terms as

$$Terms(agent, otheragent, contract) \rightarrow Observation(agent, otheragent, contract). \tag{9}$$

The buyer can be rated positively using $AdditionalPositiveRating$ and $AdditionalRatings$ to each increment the respective value via

$$\begin{aligned} &Pay(buyer, seller, value) \wedge CurrentDate \leq SellDate + 30 \\ &\rightarrow AdditionalPositiveRating(buyer, 1) \wedge AdditionalRatings(buyer, 1) \end{aligned} \tag{10}$$

and negatively as

$$\neg Pay(buyer, seller, value) \wedge CurrentDate > SellDate + 30 \\ \rightarrow AdditionalRatings(buyer, 1). \quad (11)$$

The seller may be rated similarly as

$$Give(seller, buyer, good) \wedge CurrentDate \leq SellDate + 30 \\ \rightarrow AdditionalPositiveRating(seller, 1) \wedge AdditionalRating(seller, 1) \quad (12)$$

and

$$\neg Give(seller, buyer, good) \wedge CurrentDate > SellDate + 30 \\ \rightarrow AdditionalRating(seller, 1). \quad (13)$$

A simple buyer agent might just choose the highest rated seller as

$$\exists s(s \in Sellers) \wedge \forall t(t \in Sellers) \\ PositiveRating(s)/NumRatings(s) \\ \geq PositiveRating(t)/NumRatings(t) \\ \rightarrow Terms(buyer, s, contract). \quad (14)$$

However, this simplified rating system does not take into account strategy, which we discuss next.

## 3 Signaling Versus Sanctioning

The game-theoretic designations of signaling and sanctioning games are relevant to trust and reputation systems because they address the key mechanism of whether an agent must decide who to choose or how to act [4, 10]. In this section, we propose a way of determining the influence of signaling versus sanctioning and how these properties affect the design of a trust or reputation system, eventually connecting it back to our life cycle model.

In a signaling setting, agents have private information that they may use to their advantage. The asymmetric information can be used strategically to cause adverse selection, where agents perform transactions with agents they believe to be desirable but end up with an undesirable interaction. An example of a signalling situation is where agents are purchasing mass-produced products and deciding whether to buy the product from one manufacturer or another based on quality, price, and features. In this case, agents signal to each other what they believe about other agents (specifically, the manufacturers). Statistical and probabilistic measures are most effective at measuring agents' behaviors in the signaling setting.

Sanctioning mechanisms are useful in cases of moral hazard. Moral hazard occurs when agents' utilities are uncorrelated, meaning that one agent's gain may yield another's loss, and one agent can directly exercise control over another's utility. A purchase where a buyer pays the seller and then the seller has the option of not sending the product to the buyer is an example case of moral hazard. If the seller will not be sanctioned for its behavior and will have no future relations with the buyer, then it has no incentive to send the product. Sanctioning must be credible for the agents involved to be successful, and may be performed by the agent affected by refusing future transactions, or by other agents policing the system. Modeling behavior in a sanctioning environment with rational environments means employing game theory techniques to find Nash equilibria.

As we remarked above, many real-world situations do not fall cleanly into either signaling or sanctioning situations. An agent may have some control over the quality of its products, but it is rarely impossible for an agent to make any changes to quality (pure adverse selection) or for an agent to have perfect control over quality (pure moral hazard). This distinction is blurred further by agents having differing levels of patience that influence their strategic behavior [7, 18] and also by the blurred distinction of whether an observation was intentionally communicated [3]. The amount of sanctioning comes down to how much explicit control an agent has over its communications, and also intent, which may be subtle.

In broad terms, we can distinguish two varieties of trust that apply in many computational settings with intelligent agents. We abstract the terms *Competence* and *Integrity*, as described by Smith and DesJardins [18], into

**Capabilities,** which are what an agent *can* do, and
**Preferences,** which are what an agent *will* do.

From these definitions, it is clear to see that when agents want to determine which other agents have capabilities, they need a signaling system which looks into what the agents have done before. When agents want to determine another agent's preferences and ensure that the agent will perform a desirable behavior in the future when it has the choice, then they need a sanctioning system. This is consistent with the notions of **reactive** and **anticipatory** coordination [2].

To examine the role of signaling versus sanctioning on reputation systems, it is instructive to consider three interrelated terms—trust, trustworthiness, and reputation—that are used in nonstandardized ways in the literature. We begin from basic definitions in order to capture the general intuitions about them.

**Trust** is an agent's assessment of another party along some dimension of goodness leading to expected outcomes.

**Trustworthiness** is how good a party is in objective terms. In other words, this is a measure of how worthy it is to be trusted.

**Reputation** is the general belief (among the agents in a society or community) about an agent.

Specifically, Alice may or may not trust Bob for possessing desirable attributes (these could be capabilities, resources, bandwidth, and such). Alternatively, Alice may or may not trust Bob for having his preferences aligned with hers or rather for having his preferences aligned with hers under a particular incentive mechanism. Bob may or may not be worthy of any trust Alice may place in him. Bob may or may not have a reputation for being trustworthy in the specified ways. And such a reputation may or may not be well earned.

Reputation and trust therefore can be fit into our dual categorization. Reputation involves what an agent is, as measured from its past; an agent has a reputation of having some attribute or capability, and so a reputation system in this sense is a signaling system. Trust is concerned with what an agent will do in a future situation, which concerns the agent's preferences and must be handled by a sanctioning system. However, as trust and reputation have other connotations in specific domains, such as emotion, we will maintain the distinction using the terms signaling and sanctioning.

### 3.1 Measuring Influence of Signaling and Sanctioning

Consider agents $A$ and $B$ that have strongly typed behavior, meaning that they will always behave almost the same way regardless of the situation (e.g., by offering products of some specific quality). An example of such an agent is one that controls a high-volume web service with specific offerings and finite bandwidth with little autonomy and business logic. Consider an agent $C$ that is deciding which agent to interact with between $A$ and $B$. If $C$ chooses $A$, then $C$ will receive some benefit (or loss) of utility, $b_A$. If $C$ chooses $B$, then $C$ would receive a change in utility of $b_B$. Since the agents are strongly typed, $C$'s behavior other than choosing $A$ or $B$ will not make much difference. To maximize utility, $C$ should use statistics to measure $A$ and $B$'s attributes.

Conversely, consider that agents $A$ and $B$ are rational, have full and precise control over each of their actions, and may change their behavior without any switching costs. An example of these agents would be low-volume reseller agents that have sufficient supply of substitutable products. In this case, whether $C$ chooses $A$ and $B$ matters little to $C$'s utility. Instead, $C$'s choices in negotiation and behavior with respect to $A$ or $B$ dominates $C$'s change in utility. Finding an optimal interaction strategy is how $C$ can maximize its utility.

If we write the benefit $C$ will gain with behavior $x$ when choosing agent $A$ as $b_{A,x}$, then magnitude difference of utility change between these choosing $A$ and $B$ while $C$ maintains consistent behavior is $|b_{A,x} - b_{B,x}|$. Using $C$'s ideal behavior, this can be written as $\max_x |b_{A,x} - b_{B,x}|$. When evaluated against all agents available for interaction, $S$, agent $C$'s value of the utility difference between two agents, $d_{\text{selection}}(C)$, can be written in terms of the expected rate of interaction between $C$ and another agent $A$ as $r_{A,C}$, as

$$d_{\text{selection}}(C) = \frac{1}{\sum_{A \in S} r_{A,C} + r_{C,A}} \cdot \sum_{A \in S} \sum_{B \in S} \max_{x \in H} |r_{A,C} \cdot b_{A,x} - r_{B,C} \cdot b_{B,x}|. \quad (15)$$

The normalizing term $\frac{1}{\sum_{A \in S} r_{A,C} + r_{C,A}}$ represents the reciprocal of the total interaction rate. Similarly, we may write the expected value of the utility difference between any two behaviors, $d_{\text{strategy}}(C)$, of the set of all behaviors in $H$ across all agents, as

$$d_{\text{strategy}}(C) = \frac{1}{\sum_{A \in S} r_{A,C}} \cdot \sum_{A \in S} \max_{x \in H, y \in H} |r_{A,C} \cdot b_{A,x} - r_{A,C} \cdot b_{A,y}|. \quad (16)$$

| Seller Agent | Refurb. Value | Refurb. Market Price | Unrefurb. Price | Refurb. Cost |
|---|---|---|---|---|
| $A$ | \$500 | \$400 | \$200 | \$150 |
| $B$ | \$490 | \$350 | \$250 | \$80 |

**Table 1.** Online auction refurbished laptop example data.

As $d_{\text{selection}}$ measures the impact of an agent's type and $d_{\text{strategy}}$ measures the impact of an agent's strategy, we can use these values to determine the impact of signaling and sanctioning on a multiagent interaction mechanism. In aggregation, we express the expected value of each of the values across all agents as $E(d_{\text{selection}})$ and $E(d_{\text{strategy}})$ respectively. The fraction of agents' total utility in a system that is governed by signaling, $i_{\text{signaling}}$ can be represented as

$$i_{\text{signaling}} = \frac{E(d_{\text{selection}})}{E(d_{\text{selection}}) + E(d_{\text{strategy}})}. \tag{17}$$

The fraction of utility governed by sanctioning, $i_{\text{sanctioning}}$, can be represented as

$$i_{\text{sanctioning}} = \frac{E(d_{\text{strategy}})}{E(d_{\text{selection}}) + E(d_{\text{strategy}})}, \tag{18}$$

with $i_{\text{signaling}} + i_{\text{sanctioning}} = 1$.

### 3.2 Example: Online Auction Representation

We reuse our general interaction model from Section 2.3 to show an example of applying our signaling versus sanctioning measure. Suppose agents are participating in online market for refurbished laptops outlined in Table 1.

A buyer agent, $C$, values its own utility of the refurbished laptop from $A$ at \$500 and the refurbished laptop from $B$ at \$490. It needs to decide whether to buy from $A$ or $B$ for the market price of \$400 or \$350 respectively. It costs $A$ \$150 to refurbish its laptop that it bought unrefurbished at \$200, and costs $B$ \$80 to refurbish the laptop it purchased at \$250. Both $A$ and $B$ are claiming that the laptop on sale is refurbished, but $C$ does not know for sure.

First, we investigate the case of selection. Agent $C$ can select to buy from $A$ or $B$, but $A$ and $B$ have no choice in the matter because of the online auction format. The rates of interaction from $A$'s perspective are $r_{A,A} = 0$, $r_{A,B} = 0$, $r_{A,C} = 1$, and $B$ is analogous. The rates from $C$'s perspective are $r_{C,A} = 1$, $r_{C,B} = 1$, $r_{C,C} = 0$.

First we evaluate $d_{selection}(C)$. Agent $A$ only can interact with $C$, and the maximum profit $C$ could make while still providing a laptop is \$200. Therefore, $d_{selection}(A) = \frac{1}{2} \cdot (|(\$400 - \$200) - \$0| + |\$0 - \$400 - \$200)|) = \$200$. Similarly, $d_{selection}(B) = \$100$. To compute this value for agent $C$, we must first evaluate which strategy yields the greatest difference between choosing $A$ or $B$. When the seller performs the refurbishment, $C$'s difference in utility between choosing seller $A$ and $B$ is $|(\$500 - \$400) - (\$490 - \$350)| = \$40$. When the seller does not perform the refurbishment, the difference becomes $|(\$200 - \$400) - (\$250 - \$350)| = \$100$. As the rates of interaction are symmetric, the larger of these two yields $d_{selection}(C) = \$100$. The combined expected value of the difference of selection across all three agents is $E(d_{selection}) = (\$200 + \$100 + \$100) \approx \$133$.

Next we investigate the case of sanctioning. Beginning with $A$, we find $d_{\text{strategy}}(A) = 1 \cdot |(\$400 - \$200 - \$150) - (\$400 - \$200)| = \$150$, which is the cost of refurbishing the laptop, and accordingly $d_{\text{strategy}}(B) = \$80$. To find $d_{\text{strategy}}(C)$, we also examine the sellers' behavior. If $A$ does not refurbish the before shipping laptop, but instead delivers a broken laptop, then $C$ regains only \$200 from selling the laptop at the unrefurbished price and loses its \$400 payment. Applying this evaluation with both $A$ and $B$, $d_{\text{strategy}}(C) = \frac{1}{2} \cdot \big( |(\$500 - \$400) - (\$200 - \$400)| + |(\$490 - \$350) - (\$250 - \$350)| \big) = \$270$. Putting the three of these agents' results together, we obtain $E(d_{\text{strategy}}) = (\$150 + \$80 + \$270)\,/3 \approx \$167$.

The system has $i_{\text{signaling}} = \frac{\$133}{\$133+\$166} \approx .44$ and $i_{\text{sanctioning}} = \frac{\$166}{\$133+\$166} \approx .66$. An effective reputation system for this system should emphasize sanctioning mechanisms slightly over signaling mechanisms.

## 4 Attacks on Trust and Reputation Systems

Given our logic model to represent a trust or reputation system, we can use these formalisms to give systematic treatment of types of attack or exploits for each type of action.

**discard identity:** An agent may discard its identity to remove a bad reputation and potentially acquire a new one [6], which can be expressed as $\neg ID(agent, id1) \wedge Util(agent, discardCost) \wedge ID(agent, id2) \wedge Util(agent, acquireCost)$. At a higher level, an agent could discard an identity which is not lost in anonymity, but rather used to frame another agent as a threat, for blackmail, to remove a competitor, or for other forms of sanctioning.

**acquire identity:** Sybil attacks occur when one agent creates multiple identities in order to manipulate a reputation or other aggregation system [13], expressed as $ID(agent, id2) \wedge Util(agent, acquireCost)$. Such attacks may directly influence reputation or flood out other behavior.

**select agent, negotiate:** An agent may select another agent that is known to be in a weak position or that is easy to manipulate or exploit, offer terms in negotiation without intent to fulfill them or with intent to deceive or harm the other agent, or demand a commitment by threatening to harm the other agent if not fulfilled. In subsequent negotiations, an agent may mislead another agent that a previous transaction was problematic and that reparations are needed to continue the relationship. Agents can also be selected against for sanctioning purposes.

**transaction:** An agent may not fulfill a commitment at all, fulfill it only part way or of lesser quality than expected, or provide something unexpected [17].

**update & communicate beliefs:** This transition is particularly rich in terms of the numbers and types of attacks, and can range from coordination with other transitions (e.g., acquiring identities in a Sybil attack) to simply communicating [5]. An agent can lie about another agent's performance for positive or negative reciprocity.

Revisiting Figure 1, the different types of attacks can be categorized by whether or not they can be primarily addressed by signaling or sanctioning matter. Selecting agents and updating and communicating beliefs are two of the interactions in the life cycle that can be exploited in the most complex strategic manners, but also map directly to signalling systems.

## 5 Discussion

In this paper, we have examine an architectural view of the trust and reputation life cycle. The motivation of this paper was not to introduce any new mechanisms of trust and reputation, but rather to take a step toward formalizing the taxonomy and structure of trust and reputation. Our model aids in the discussion of systems' mechanisms, as it presents a broadly applicable view that can be used in conjunction with other taxonomies of reputation systems [14, 1, 15, 9] and taxonomies of attacks [12]. Further, we offer an indication as to the extent that a given interaction system is affected by both signalling and sanctioning, which is useful in determining what kind of a reputation or trust system should be deployed. Our work unifies reputation systems, trust systems, and related game theory under a common architectural framework.

Applying our measure of the effect of signaling versus sanctioning to a live system requires some judgement. For example, if the strategy which offers the worst utility for a given situation is something that no rational agent would do, then it is best left out. In general, only those actions that lie on the Pareto frontier of utility or on a Nash equilibrium (or approximate Nash equilibrium) should be considered. The actual attributes of the agents, including preferences, utilities, and capabilities may also be unknown to a system designer, so estimates may often need to be substituted.

A key theme in our model is the relative autonomy of the agents involved. If the agents behave in a fixed manner that is largely independent of the other agents' strategies, then are best measured by a signaling system. An agent's autonomy is further reflected by the level of bounded rationality and information available to an agent in the system.

Future work involves investigating time preferences of agents and combinations of actions. Though our model is high level, some strategies require involvement of several distinct pieces of the model. Further future work involves deepening the connection between our logic framework with the signaling and sanctioning measures to give more prescriptive results.

## References

1. D. Artz and Y. Gil. A survey of trust in computer science and the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5:58–71, 2007.
2. C. Castelfranchi. Modelling social action for AI agents. *Artificial Intelligence*, 103(1-2):157–182, 1998.
3. C. Castelfranchi. Silent agents: From observation to tacit communication. *Lecture Notes in Computer Science*, 4140:98–107, 2006.

4. C. Dellarocas. Reputation mechanisms. In T. Hendershott, editor, *Economics and Information Systems*, volume 1 of *Handbooks in Information Systems*, chapter 13, pages 629–660. Elsevier Science, 2006.
5. J. Farrell and M. Rabin. Cheap talk. *The Journal of Economic Perspectives*, 10(3):103–118, 1996.
6. M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica. Free-riding and whitewashing in peer-to-peer systems. *IEEE Journal on Selected Areas in Communications*, 24(5):1010–1019, 2006.
7. C. J. Hazard. ¿Por favor? Favor reciprocation when agents have private discounting. In *AAAI Workshop on Coordination, Organizations, Institutions and Norms (COIN)*, pages 9–16, Chicago, Illinois, July 2008.
8. 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 MultiAgent Systems*, 13(2):119–154, 2006.
9. A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, March 2007.
10. R. Jurca and B. Faltings. Obtaining reliable feedback for sanctioning reputation mechanisms. *Journal of Artificial Intelligence Research*, 29:391–419, August 2007.
11. S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in P2P networks. In *Proceedings of the 12th International Conference on World Wide Web*, pages 640–651, Budapest, Hungary, 2003.
12. R. Kerr and R. Cohen. Smart cheaters do prosper: Defeating trust and reputation systems. In *Proceedings of the eighth international conference on autonomous agents and multiagent systems*, pages 993–1000, 2009.
13. J. Newsome, E. Shi, D. Song, and A. Perrig. The sybil attack in sensor networks: analysis & defenses. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 259–268, New York, NY, USA, 2004. ACM.
14. S. D. Ramchurn, D. Huynh, and N. R. Jennings. Trust in multi-agent systems. *The Knowledge Engineering Review*, 19:1–25, 2004.
15. J. Sabater and C. Sierra. Review on computational trust and reputation models. *Artificial Intelligence Review*, 24(1):33–60, September 2005.
16. A. Salehi-Abari and T. White. Towards con-resistant trust models for distributed agent systems. In *International Joint Conference on Artificial Intelligence*, pages 272–277, 2009.
17. M. P. Singh. An ontology for commitments in multiagent systems: Toward a unification of normative concepts. *Artificial Intelligence and Law*, 7(1):97–113, March 1999.
18. M. J. Smith and M. desJardins. Learning to trust in the competence and commitment of agents. *Autonomous Agents and Multi-Agent Systems*, 18(1):36–82, February 2009.
19. J. D. Sonnek and J. B. Weissman. A quantitative comparison of reputation systems in the grid. In *Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing*, pages 242–249, 2005.
20. M. Srivatsa, L. Xiong, and L. Liu. Trustguard: Countering vulnerabilities in reputation management for decentralized overlay networks. In *Proceedings of the 14th International Conference on World Wide Web*, pages 422–431, 2005.

# Modeling Virtual Footprints

Rajiv Kadaba, Suratna Budalakoti, David DeAngelis, and K. Suzanne Barber

The Laboratory for Intelligent Processes and Systems
The University of Texas at Austin
University Station C5000, ACE 5.124
Austin, Texas, 78712-0321 USA
{kadaba,sbudalakoti,dave,barber}@lips.utexas.edu

**Abstract.** Entities interacting on the web establish their identity by creating virtual personas. This research models identity using the *Entity-Persona Model* which is a semantically annotated social network inferred from the persistent traces of interaction between personas on the web. A *Persona Mapping Algorithm* is proposed which compares the local views of personas in their social network referred to as their *Virtual Signatures*, for structural and semantic similarity. The semantics of the social network of the *Entity-Persona Model* is modeled by a vector space model of the text associated with the personas in the network, which allows efficient comparison of their *Virtual Signatures*. This enables all the publicly accessible personas of an entity to be identified on the scale of the web. This research enables an agent to identify a single entity using multiple personas on different networks. The agent is able to increase the trustworthiness of on-line interactions by establishing the identity of entities operating under multiple personas. Consequently, reputation measures based on on-line interactions with multiple personas can be aggregated and resolved to the true singular identity.

**Keywords:** trust, social networks, identity management, virtual signatures

## 1 Introduction

The way that an individual's identity is created and experienced is fundamentally different in the virtual world. The basic cues used to uniquely identify individuals in the real world are missing, making the association between an entity and its identity ambiguous [14]. This research creates a model of the virtual world which dispels this ambiguity, allowing the virtual personas created by an entity to be linked together. Informally, a virtual persona is a name and its associated attributes, which an entity uses to communicate with other personas.

The virtual world in the context of this research refers collectively to the various explicit or inferred social networks on the web. Examples of explicit social networks are websites such as Facebook, Orkut, MySpace, and LinkedIn. Social networks can be inferred from the digital traces of interaction between entities, or individuals, on the internet, such as in the Blogosphere [9], Online

Discussion Forums, Knowledge sharing sites, IRC Logs and the co-occurrence of names in the large amount of textual data on the internet [7]. In an explicit social network there exists a framework by which entities can specify to whom they are related and the context of this relationship. Access to explicit networks is generally controlled [6] because of the privacy concerns of its participants [1].

Inferred social networks lack the privacy mechanisms of explicit networks as its users assume they are as anonymous as they wish to be. The work in this paper counters this assumption since users must engage in information rich interactions in order to provide value to the framework. The establishment of the reputation of an entity's virtual persona within the framework is an important motivating factor for its consistent use, as others use reputation to assess the reliability of information associated with the persona [5]. Every new persona created will need to establish its reputation within its social network which requires time and effort. This penalty associated with creating a persona which is capable of meaningful interaction makes a persona valuable. Virtual personas with erratic interactions are not worth detecting as they have little value.

Search engines treat personal names and pseudonyms as keywords, giving virtual personas the same status as ordinary text. Queries for people's names only find occurrences with verbatim matches to the query text while they may have interacted extensively using various personas. The model proposed in this paper can be used to find more accurate results for the information associated with an individual available publicly on the web. Augmenting web search with the ability to link entities and their personas can be perceived as an attack on an individual's privacy, as information which may have been exchanged with the expectation of anonymity granted by a virtual persona is now linked back to its progenitor. Conversely this research can also contribute to an individual's ability to safeguard their privacy and protect their identity from theft. As the concept of identity in the virtual world is formalized and an upper bound on the ability of a determined adversarial agent is found, techniques to remain anonymous in spite of sophisticated statistical tools can be developed. Further, software agents who are capable of warning users of the unintended inferences which can be made with the data they publish may also be possible.

Another application of this research is in the task of anti-aliasing in social networks. Anti-aliasing [11] is the task of identifying when a single user has multiple aliases in a social network. This is important in trust networks, as a distrusted user or set of users, after being removed from a network (by a moderator, for example), can insert themselves into a network at a later point in time with a pseudonym. Agents capable of continuously associating personas with the singular identity of an entity will offer increased assurance of entity reputations based on interactions of associated personas. Consequently, agents can assess trustworthiness of individual personas and relate those trustworthiness assessments to the singular identity associated with those personas. This research offers the groundwork for establishing the connection between the multiple personas and a singular entity identity.

## 2 Related Work

The privacy of individuals, referenced in social network data released to researchers, application developers, and marketing organizations is ostensibly protected by anonymizing the social graph, as their goal is to make inferences about the aggregated data not specific individuals. Privacy preserving data mining is a research area in which data sets are modified and algorithms developed which do not compromise privacy [16]. Approaches to the reverse task of de-anonymization are usually based on unique subgraphs in the anonymized social network, and are classified into active and passive attacks [3]. Active attacks are attacks in which nodes that form a unique subgraph are inserted into the graph before it is anonymized (for example, before being released to the public for research of other purposes). As this subgraph is known a priori it can be used to identify other nodes in the released graph. Passive attacks [10] are similar, however no nodes are inserted, instead a small group of nodes collude to generate a subgraph which is later used to re-identify adjacent nodes. De-anonymization of individuals in an anonymized data set is equivalent to linking personas in two different social networks as the attackers background information is also a social network. These techniques de-anonymize nodes using structural properties of graphs and are not designed to target specific nodes. They also rely on the fact that both networks have many nodes in common. Purely structural equivalence techniques break down if the neighborhood of an individual node changes drastically. The work presented here uses structural information in addition to persona content for de-anonymization.

In contrast to graph-based approaches, Novak et al. [11] use a content-based approach where they reconcile online personas to unique users by clustering the content associated with the personas, such that each cluster represents a unique user. The data is derived from an online discussion board by only considering text associated with a persona independent of relation information. Similar to the approach used in this paper, two sets of personas are synthetically created by random division. Random division allows words from specific topics to appear in every division which will not occur in real data, making the entity disambiguation results flawed. Algorithms must be robust with respect to text originating from very specific topics which are never repeated. Temporal division as used here addresses this problem.

Jin et al. [7] extract social networks from the web using hypertext data retrieved from search engine queries. The nodes in the network are named entities which are known a priori. Edges are inferred using heuristics from co-occurrence of names, and the type of relationship the edge represents is determined from the query text. Queries consist of the entity names and the type of relationship. Staddon et al. [13] have leveraged web search to determine unintended inferences which can be drawn from data published on the web. Keywords are extracted from data intended to be published using TF-IDF[1] and are used to construct queries to search engines. New keywords in the results of these queries not present

[1] Term Frequency - Inverse Document Frequency

in the original keyword set represent inferences which can be drawn from the web. Both these techniques only find information associated with a specific name i.e. they assume an individual has only one persona on the internet. Although the results are interesting they bring little insight to the true nature of identity on the web.

## 3 Modeling Entities and Personas

Entities communicate on the web using identifiers unique within a framework making the identifier - framework combination also unique. At internet scope the identifier can be an IP or email address depending on the protocol, or a username in the scope of a Web 2.0 application. The identity of an entity possessing an identifier is characterized by the set of other identifiers it has communicated with and the content of this communication, collectively referred to as its virtual signature.

### 3.1 Model and Definitions

**Definition 1.** *An entity $\xi$ is something capable of independent interaction, which can be uniquely identified in the real world.*

An entity can be an individual or software agent. Individuals by default are unique as they can have only one instance in the real world. A software agent may not be unique as it is very easy to create many instances of the same agent. Therefore, all instances of the same software agent which exhibit the same behavior are considered collectively a single entity regardless of their physical location. Entities interact on the web through a framework using a persona, which is an instantiation of an entity within the framework. An entity can possess more than one persona within a given framework.

**Definition 2.** *A framework is an implementation of software and associated protocols which enables entities to interact.*

**Definition 3.** *A persona $\pi$ is a tuple $(i, d)$, where $i$ is an identifier unique within a framework and $d$ is a n-tuple of associated information and attributes which an entity uses to establish its identity and interact within a social network. For every persona there exists exactly one entity*

A social network (Definition 4) is used to model the virtual signatures of entities or personas which are nodes in the network such that two nodes are connected if they have communicated. Again, the criteria for considering that communication has occurred depends on the framework.

**Definition 4.** *A social network $S$ is a vertex and edge labeled undirected graph $G = (V, E)$, where $V$ is a set of either exclusively entities or personas and $E$ is a subset of the cartesian product $V \times V$ such that $(v, v) \not\subset E$* [2]*. Every edge*

[2] No self loops are allowed since they are meaningless in a communication graph.

*$(u, v)$, $u, v \in V$ has a label $\gamma$ and every vertex has a label $\chi$, which is arbitrary information associated with the edge or vertex.*

The *Entity-Persona Model* consists of a social network of entities $S_{\Xi}$ and at least one social network of personas $S_{\Pi}$. $S_{\Xi}$ is a real world social network of entities whose personas need to be linked. $S_{\Pi}$ is a social network of personas inferred from the records of the framework in which the entities in $S_{\Xi}$ communicate. The label of vertex $u_{\pi}$, $\chi_{\pi}$ in the graph $G_{\Pi}$ of $S_{\Pi}$ is $\pi(i, T(d))$ and the label $\gamma_{\pi_1\pi_2}$ of edge $(u_{\pi_1}, v_{\pi_1})$ between vertex $u_{\pi_1}$ labeled by $\pi_1$ and vertex $v_{\pi_1}$ labeled by $\pi_2$ is $T(d_{\pi_1} \cup d_{\pi_2})$. $T$ is an operation which reduces $d$ to its semantics which is outlined in section 3.3. Hence, the social network of personas is vertex labeled by the semantics of information generated by the personas and edge labeled by the semantics of all the information exchanged by the personas it connects.

**Definition 5.** *The virtual signature of a persona $\pi$ in the context of the* Entity-Persona Model *is the social network $S_{\Pi}$ rooted at the node of the persona.*

The identity of a persona is defined by the structure of $S_{\Pi}$ and the semantics of the graph labels. $S_{\Pi}$ is not unique to a persona as it is a social network of personas, however its local view can be unique making its virtual signature also unique.

### 3.2 Problem Specification

Linking entities and their corresponding personas can be formally expressed as finding a mapping between a set of $n$ entities $\Xi = \{\xi_1, \xi_2, ..., \xi_n\}$ which are nodes in a social network $S_{\Xi}$ and a set of $m$ personas $\Pi = \{\pi_1, \pi_2, ..., \pi_m\}$. $\Pi$ may be partitioned into $P = \{\Pi_1, \Pi_2, ..., \Pi_l\}$ a collection of $l$ subsets of $\Pi$ if the personas exist in $l$ different frameworks. Each framework is expressed as a separate social network in the model.

**Definition 6.** *A mapping $\mu$ is a binary relation on $\bigcup\limits_{i=1}^{l} S_E \times S_{\Pi_i}$, where $(\xi, \pi) \in \mu, \xi \in S_E, \pi \in S_{\Pi_i} \iff$ entity $\xi$ has created persona $\pi$.*

From the information exchanged within each framework, $P$ can be inferred. A query on the model is a set of entities $\Xi$ and a mapping $\mu$ which maps every entity in $\Xi$ to at least one persona in $P$. If no mapping between an entity and a persona is known, the query algorithm will not have an example to train on making mapping the personas of the entity impossible. The social network $S_{\Xi}$ and the mapping $\mu$ are only partially known. From this information the complete social network $S_{\Xi}$ and $\mu$ must be inferred. This can be accomplished by,

1. Selecting a entity $\xi \in \Xi$.
2. For $(\xi, \pi_{\xi})$ in mapping $\mu$, find a set $\Pi_{new}$ which contains all $\pi \in P$ that best match $\pi_{\xi}$ by some objective criteria.
3. Update mapping $\mu$ such that $\mu = \mu \bigcup\limits_{q \in \Pi_{new}} (\xi, q)$.

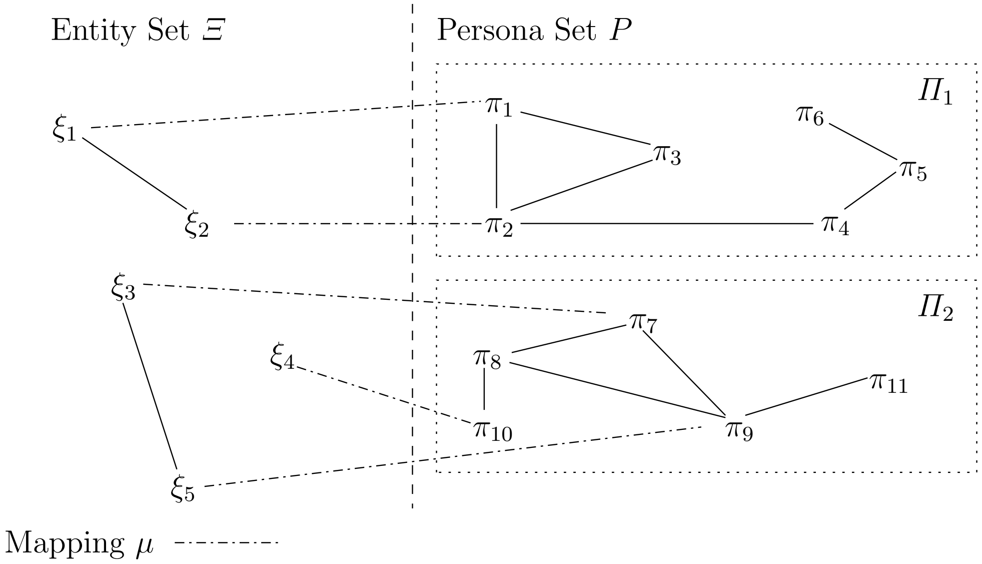


**Fig. 1.** The entity-persona model as it is initilalized.

4. Update $S_{\mathit{\Xi}}$ such that if two mapped personas have an edge between each other, their corresponding entities have an edge in $S_{\mathit{\Xi}}$.
5. Repeat.

### 3.3 Semantic Similarity

If the virtual signatures (Definition 5) of personas are to be compared, it follows that there must be a way to compare their local views of their social network. Graphs can be compared structurally using combinatorial algorithms [4] or by comparing their spectra [15]. In a social network however the nodes and edges are labeled requiring the development of the means to compare these labels.

The edge labels of a social network are the inferred semantics of the relationship the edges represent. The vertex labels are the inferred semantics of all the information generated by an entity through the persona associated with the vertex. These semantics must be short descriptions of the information they are inferred from, while maintaining as much discriminative information as possible.

The simplest technique to infer the semantics of text associated with a persona is to construct a vector of unique terms in the text and assign it to a graph element label. Labels can be compared by using the Jaccard index which measures the similarity between two sets $\chi_1$ and $\chi_2$.

$$J(\chi_1, \chi_1) = \frac{|\chi_1 \cap \chi_2|}{|\chi_1 \cup \chi_2|} \tag{1}$$

This produces a score between 0 and 1, with 1 being exactly the same. It is good at using stylometric information such as consistent misspellings to discriminate between personas but cannot capture information such as favorite words. It is particularly weak at discriminating when the number of terms in the original text is very small.

Term Frequency - Inverse Document frequency is a popular technique in information retrieval to infer which terms best represent a document in a corpus. In the case of the *Entity-Persona Model*, the document is the textual information associated with a persona and the corpus is the collection of all personas. This technique can reduce the dimensionality of the feature space to a greater extent than the previous approach as only the most important terms can be considered i.e. terms with the highest tf-idf. To compare labels $\chi_j$ and $\chi_k$ the cosine similarity between their vectors is computed,

$$\text{Cosine Similarity}_{\chi_j,\chi_k} = \frac{\sum\limits_{\forall i} tf-idf_{\chi_j,t_i} \times tf-idf_{\chi_k,t_i}}{||tf-idf_{\chi_j}|| \times ||tf-idf_{\chi_k}||} \quad (2)$$

Cosine similarity results in a score between $-1$ and 1 with 1 being exactly the same. Therefore a higher score implies that the labels being compared are similar.

### 3.4 The Persona Mapping Algorithm

The algorithm takes as input, a source graph $G_S$ of a social network of personas $S_S$, a source vertex $s \in G_S$ of a persona $\pi_s$, and a target graph $G_T$ of a social network of personas $S_T$. $S_S$ and $S_T$ can be the same social network, in this case we are looking for an entity with multiple personas in the same social network.It returns a vertex in target graph which has the maximum overlap between its virtual signature and the signature of $s$. The algorithm is a modified simultaneous *Breadth First Search* of two graphs. It also takes as inputs $\epsilon$ and $\Delta$ which help prune the search space, and $\sigma$ which is the standard deviation of a gaussian used to weigh the importance of a semantic or structural information while producing a score of similarity between two personas.

As the personas of an entity may not have the same neighbors in their social graphs, the structure of the local view of their social network can vary. Depending on the framework the personas are inferred from they may exchange information on very different topics making semantic information unreliable. If both semantic and structural information change drastically a persona may not be able to be mapped correctly, but if only one of them change the algorithm can take this into account by weighing both types of information differently using a zero mean gaussian to decide on the weight of the scores of vertices in the auxiliary graph. The larger the path length between the vertex which maps the source vertex to the candidate vertex and a vertex in the auxiliary graph, the less that vertex contributes to the score. If $s_{i,d}$ is the score of the $i^{th}$ vertex at path length $d$ then the score of the similarity of the candidate vertex to the source vertex is given by:

$$score = \sum_{d=0}^{\Delta} \frac{\sum_{\forall i} s_{i,d}}{|s_{i,d}|} \left( \frac{1}{\sqrt{2\pi}} e^{-\frac{d^2}{2}} \right) \quad (3)$$

**Algorithm 1** Persona Mapping Algorithm.

```
MAP-NODE($G_S, G_T, s, \epsilon, \sigma, \Delta$)
$Q \leftarrow \emptyset$
for each vertex $v \in V[G_T]$
    do $Q \leftarrow$ GET-VERTEX-SCORE($G_S, G_T, s, v$)
while $Q \neq \emptyset$
    do $VertexScore \leftarrow$ MAXIMUM($Q$)
       $c \leftarrow$ EXTRACT-MAX($Q$)
       $Q_S, Q_T \leftarrow$ INITILIZE-QUEUES($s, c$)
       $G_A \leftarrow$ INITILIZE-GRAPH($s, c, VertexScore$)
       while $Q_S \neq \emptyset$ and CHECK-DEPTH($\Delta$)
           do $node_1 \leftarrow$ DEQUEUE($Q_S$)
              $node_2 \leftarrow$ DEQUEUE($Q_T$)
              $AssignedEdges \leftarrow$ MAP-EDGES($node_1, node_2$)
              UPDATE-GRAPH($G_A$, $AssignedEdges$)
              $Q_S$, $Q_T \leftarrow$ UPDATE-QUEUES($Q_S$, $Q_T$, $AssignedEdges$)
       $Score \leftarrow$ GET-COMBINED-SCORE($G_A, \sigma$)
       if $LastScore - Score \geq \epsilon$
          then return $LastC$
          else $LastScore \leftarrow Score$
               $LastC \leftarrow c$
```

Subgraph isomorphism is a NP-complete problem, the *Persona Mapping Algorithm* avoids this pitfall. It builds an intersection graph between two subgraphs greedily to check for similarity making the problem tractable. The time complexity of the algorithm is $O(n^4 + n \log n)$. However, real world social networks have node degrees which follow a power law distribution [2]. Therefore the edge mapping routine will take on average $O(d^2)$, where $d$ is the average node degree of the network. The use of $\Delta$ to limit the depth of the breadth first search results in a further decrease in running time to build the auxiliary graph. Empirically a depth of 2 or 3 is sufficient to achieve a correct mapping as larger depths will include the entire graph due to the small world phenomenon [8], which in most cases is unnecessary.

Percentage of personas mapped correctly is a good measure of performance on a given data set. It is argued by [10] that this metric is flawed because nodes which are impossible to map will bias the results. However, this does not matter while measuring relative performance on a given data set. Finding a mapping between one entity and its personas is as important as the other, using measures such as node degree or centrality to give more weight to the successful mapping of more important nodes is meaningless here. For the same reason doing well on personas with relatively less information associated with them is also not considered.

# 4 Experiments

## 4.1 Outline of Experiments

This section presents experiments which validate the *Entity-Persona Model* and the *Persona Mapping Algorithm* against the Enron data set. The experiments test the robustness of the algorithm and the semantic similarity measures it uses by artificially partitioning the data set. At least 50% of the entities in the data set had their personas correctly mapped in every case.

The Enron data set was initially released by the Federal Energy Regulatory Commission as part of its Western Energy Markets investigation. The particular version used by this work has been prepared by Shetty and Adibi [12]. This data set is an appropriate test bed for this research as it is only composed of text and the link structure is easy to infer. As it is a very mature data set, it required minimal preprocessing to make it usable. Its size also makes it appropriate for research while still exhibiting many of the properties of large social networks.

The Enron data set consists of 517,431 emails from the mail accounts of 151 Enron employees between January 1998 to December 2002 with most of the email volume occurring between January 2000 and December 2001. It contains emails that originate or are sent to addresses outside those of the employees in the data set, these have not been considered in these experiments. The *Entity-Persona Model* is populated by considering every email address as a node and a edge between two nodes is added if they have exchanged at least one email. The vertex labels of the social network are inferred from an unordered collection of terms in the text of all the emails sent or received by a node. The edge labels

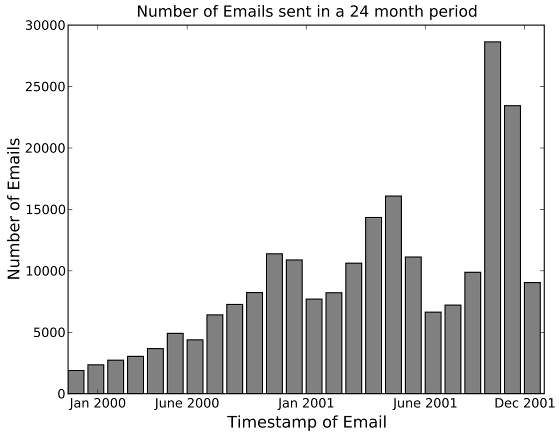


**Fig. 2.** Distribution of Enron email volume over a 2 year period.

are inferred from the common emails exchanged. Only the subject and body of the email is considered. To simulate anonymous interaction personal names and email addresses are filtered out. No stop words or stemming is used as this will not allow the stylometric features of the text to be captured. The goal of the persona mapping algorithm is to map personas unknown to it. A true test of its capabilities is its ability to map personas attempting to be anonymous. Although the web is full of this kind of data, it will be impossible to verify if the algorithm has made a correct mapping as this would require entities to volunteer this information. The next best approach is to synthetically create data sets to run the algorithm on, allowing the mappings to be easily verified. This research uses *temporal partitioning* on the Enron data set.

The data set is partitioned into eight sets based on when an email is sent as shown in table 1. The effect of the change in topics of the emails and change in graph structure can be explored. From table 2 it can be seen that each of the partitions has a different graph structure. At the local view of a node the graph changes significantly with some nodes communicating with completely new neighbors. The partition with the largest email volume is chosen to derive the source graph, this is the graph which the mapping between an entity and a persona is known. The seven other partitions are used to derive the target graphs whose personas need to be mapped.

| | | Partition by Email Date | |
|---|---|---|---|
| Graph Type | Graph Name | Start Date | End Date |
| Source Graph | $G_S$ | $1^{st}$ October 2001 | $31^{st}$ December 2001 |
| | $G_{T1}$ | $1^{st}$ July 2001 | $31^{st}$ August 2001 |
| | $G_{T2}$ | $1^{st}$ April 2001 | $30^{th}$ June 2001 |
| | $G_{T3}$ | $1^{st}$ January 2001 | $31^{st}$ March 2001 |
| Target Graph | $G_{T4}$ | $1^{st}$ October 2000 | $31^{st}$ December 2000 |
| | $G_{T5}$ | $1^{st}$ July 2000 | $31^{st}$ August 2000 |
| | $G_{T6}$ | $1^{st}$ April 2000 | $30^{th}$ June 2000 |
| | $G_{T7}$ | $1^{st}$ January 2000 | $31^{st}$ March 2000 |

**Table 1.** Partitioning the Enron Data Set.

### 4.2 Comparison of Semantic Similarity Techniques

The semantic similarity techniques presented in section 3.3 which are used by the *Persona Mapping Algorithm* are compared here. The experiment measures the percentage of personas successfully mapped between the source and seven target data sets. The success of these techniques depends on term usage by the entities in the the data sets. An analysis of the dataset showed that the number of unique words used by an entity is weakly correlated with the volume of communication.

| Statistic | $G_S$ | $G_{T1}$ | $G_{T2}$ | $G_{T3}$ | $G_{T4}$ | $G_{T5}$ | $G_{T6}$ | $G_{T7}$ |
|---|---|---|---|---|---|---|---|---|
| Nodes | 139 | 137 | 140 | 109 | 107 | 93 | 79 | 58 |
| Node Degree (Max) | 63 | 54 | 63 | 24 | 22 | 18 | 11 | 14 |
| Node Degree (Avg) | 11 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |
| Edges | 825 | 616 | 531 | 345 | 328 | 234 | 127 | 88 |
| Email Volume (Avg) | 531 | 466 | 418 | 457 | 448 | 432 | 306 | 338 |
| Email Volume (Med) | 257 | 192 | 116 | 66 | 58 | 24 | 2 | 0 |

**Table 2.** Graph Statistics

This is the basis of the success of using textual information to find mappings as an entity does not have to have communicated extensively for there to be enough information to uniquely identify it.

The algorithm was run with $\sigma$ set as 0.6, $\Delta$ as 1 and $\epsilon$ as 0.002. These settings give more weight to the score of the mapping of the candidate vertex to the source and less to the scores of their common neighbors. This can interpreted as the algorithm has less confidence in that entities use their personas to communicate with the same people and more confidence in that exchanging similar information across all their personas. A smaller $\Delta$ implies the algorithm only looks at the immediate neighbors of the personas for structural congruence.

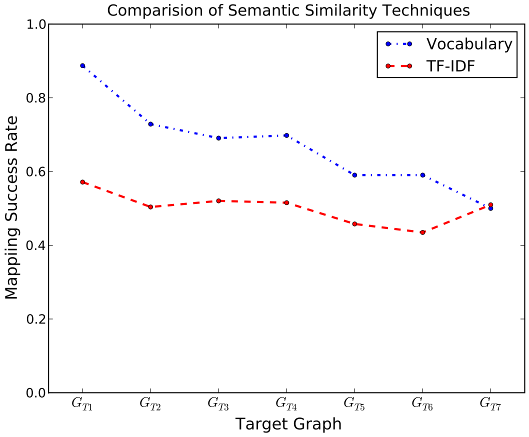


**Fig. 3.** Comparison of Semantic Similarity Techniques.

The Jaccard index to compare persona vocabulary was the more successful semantic similarity measure. It was also the least consistent in producing successful mappings as the time between which an entity used its persona increased. This implies it is the least robust to changes in specific topics of communication.

| Semantic Similarity Technique | Average Mapping Time |
|---|---|
| Jaccard Index | $2.18s$ |
| TF-IDF | $25.25s$ |

**Table 3.** Comparison of Time Required to Map

It is also the least computationally intensive technique to transform raw text into a representation which can be compared. Only one pass was required to populate the vocabulary vectors which label the graph.

TF-IDF performs more consistently although with a lower correct mapping rate. As term frequency is taken into account, it is less sensitive to rarely used terms. It correctly mapped the same subset of entities through all the data sets. It requires at least two passes over the email text to form the TF-IDF vector and is very slow at finding a mapping because the length of its vector is the size of the vocabulary of all the personas.

## 5 Conclusions

This research formally defines the problem of identifying the personas of entities on the web and proposes a solution to identify the two personas of an entity which exist in two separate social networks. This solution can be easily extended to the more general problem of multiple social networks and multiple personas. This research aims to answer the question: "Who am I interacting with?" or in other words, "Can I trust the identity presented as a true representation of the entitys identity?" This research takes a first step by offering a measurement of similarity among digital personas to resolve the true identities. Consequently, increased assurance of digital identities will allow for increased trustworthiness for on-line interactions with unknown entities (e.g. email, social networks, e-commerce, etc.).

The *Entity-Persona Model* is a formal model of the social graph and the *Persona Mapping Algorithm* operates on this model to produce mappings between entities and their personas. The *Entity-Persona Model* was populated using the Enron data set and the performance of the proposed algorithm was studied. Although the algorithm was successful in correctly mapping the entities in the data set to their personas, it was not consistent in its performance. The vocabulary, TF-IDF, and topic distribution approaches for semantic similarity compared in this research differ in their robustness to change in topics of information exchanged and the ability to capture stylometric information. These approaches can be improved or combined to perform consistently across all personas.

The question of what information generated by an entity is necessary and sufficient to uniquely identify it is the hardest to answer. It is apparent that the relative information entropy between the digital signatures of personas to be compared must differ by at least one bit in order to discriminate between them, which gives a definite lower bound for necessary information and can occur in a fully connected social network. This lower bound however is not very useful in a real world situation as every virtual persona will have a unique signature. An empirical bound on the sufficiency of information can be found if all personas in the data set have been mapped correctly. However it will be impossible to know if all the personas have been mapped correctly as the purpose of the algorithm is to find unknown mappings and there is no ground truth to reference. User feedback can be used to learn how much information is sufficient by asking the user if he is satisfied with the results of the mapping, and then adjusting the amount of information considered based on this feedback. The danger of additional information adding more noise than signal must be considered while using such techniques.

## References

1. Ro Acquisti and Ralph Gross. Imagined communities: Awareness, information sharing, and privacy on the facebook. In *In 6th Workshop on Privacy Enhancing Technologies*, pages 36–58, 2006.
2. Lada A. Adamic, Rajan M. Lukose, Amit R. Puniyani, and Bernardo A. Huberman. Search in power-law networks. *Phys. Rev. E*, 64(4):046135, Sep 2001.
3. Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 181–190, New York, NY, USA, 2007. ACM.
4. L.P. Cordella, P. Foggia, C. Sansone, and M. Vento. A (sub)graph isomorphism algorithm for matching large graphs. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(10):1367–1372, Oct. 2004.
5. Judith Donath. *Identity and Deception in the Virtual Community.* Routledge, London, 1999.
6. James Grimmelmann. Facebook and the social dynamics of privacy. Draft article, August 2008.
7. Yingzi Jin, Yutaka Matsuo, and Mitsuru Ishizuka. Extracting social networks among various entities on the web. In *ESWC '07: Proceedings of the 4th European conference on The Semantic Web*, pages 251–266, Berlin, Heidelberg, 2007. Springer-Verlag.
8. Jon Kleinberg. The small-world phenomenon: An algorithmic perspective. In *in Proceedings of the 32nd ACM Symposium on Theory of Computing*, pages 163–170, 2000.
9. Ravi Kumar, Jasmine Novak, Prabhakar Raghavan, and Andrew Tomkins. Structure and evolution of blogspace. *Commun. ACM*, 47(12):35–39, 2004.
10. Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. Mar 2009.
11. Jasmine Novak, Prabhakar Raghavan, and Andrew Tomkins. Anti-aliasing on the web. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 30–39, New York, NY, USA, 2004. ACM.
12. Jitesh Shetty and Jafar Adibi. The enron email dataset database schema and brief statistical report. Technical report, Information Sciences Institute, 2004.
13. Jessica Staddon, Philippe Golle, and Bryce Zimny. Web-based inference detection. In *SS'07: Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, pages 1–16, Berkeley, CA, USA, 2007. USENIX Association.

14. Sherry Turkle. *Life on the Screen: Identity in the Age of the Internet*. Simon & Schuster, September 1997.
15. S. Umeyama. An eigendecomposition approach to weighted graph matching problems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 10(5):695–703, Sep 1988.
16. Vassilios S. Verykios, Elisa Bertino, Igor Nai Fovino, Loredana Parasiliti Provenza, Yucel Saygin, and Yannis Theodoridis. State-of-the-art in privacy preserving data mining. *SIGMOD Record*, 33:2004, 2004.

# Engineering Trust Alignment: a First Approach

Andrew Koster, Jordi Sabater-Mir, and Marco Schorlemmer

IIIA, Artificial Intelligence Research Institute
CSIC, Spanish National Research Council
Bellaterra, Spain
{andrew, jsabater, marco}@iiia.csic.es

**Abstract.** In open multi-agent systems trust models are an important tool for agents to achieve effective interactions. However, in these kinds of open systems, the agents do not necessarily use the same, or even similar, trust models, leading to semantic differences between trust evaluations in the different agents. Hence, to successfully use communicated trust evaluations, the agents need to align their trust models. We explicate that currently proposed solutions, such as common ontologies or ontology alignment methods, lead to additional problems and propose a novel approach. We show how the trust alignment can be formed by considering the interactions agents share. We describe our implementation of a method, which uses inductive learning algorithms, to accomplish this alignment and test it in an example scenario.

## 1 Introduction

In open multi-agent systems, trust and reputation models are considered an important feature in managing the social environment agents are immersed in. One of the benefits offered by using trust is that agents can communicate their trust evaluations to each other, thus warning other agents for fraudulent agents or helping each other to select good interaction partners. This communication, however, becomes problematic if the different agents use diverse models of trust, as is a very real possibility in a heterogeneous environment. In this case, trust may mean something different to both agents. These agents need to align their definitions of trust, before the communication becomes meaningful to them.

Trust in computational systems, as in human environments, cannot be seen as independent from the social interactions on which it is based. Each agent computes its trust evaluations of the different target agents in the system, based on some, possibly partial, observation of that agent's interactions. These trust evaluations are computed by an agent's trust model and thus, if the agents use different trust models, then the trust evaluations may be different, despite being based on the same interactions. Additionally we argue that this can be the case even if the agents use the same computational trust model. Some state of the art trust models are based on cognitive principles [1, 2] and take an agent's beliefs and goals into account when computing a trust evaluation. While, in these cases, the computational model is the same, agents with different beliefs and goals will have different trust evaluations given exactly the same information. We see that also in these cases it is therefore important to align the trust models.

The interactions trust is based on depend largely on the purpose of the multi-agent system and the types of agents in it. They can be as diverse as air traffic agents optimizing a landing schedule, personal agents buying a bicycle on eBay or agents communicating trust evaluations. The developers of the system generally develop an ontology to facilitate communication about these interactions and the domain in general. This should facilitate the communication about the interactions, however the trust evaluations are not based on public information alone. An agent may have its own personal observations of an interaction. In the case of an eBay auction, the seller for instance has information about all bids received, while the bidder only has information about his own bids. Additionally agents may associate different subjective observations with the interaction. For instance, the seller may not be satisfied with the transaction, because he had to sell at a loss. This type of information is private and often just as subjective as the trust evaluation itself. In the eBay example it is no easier to communicate the meaning of satisfaction than of the trustworthiness supported by that satisfaction. This difference in observations complicates the matter of aligning trust models, however we postulate that there is always some amount of shared information. At the very least, there is shared information that an interaction took place. Our approach uses these shared interactions as building blocks for a trust alignment.

So far, communication about trust evaluations has been tackled by defining common ontologies for trust [3, 4], however in practice these ontologies do not have the support of many of the different trust methodologies in development. An ontology alignment service is presented in [5], but it requires a translation of all specific trust model ontologies into a general ontology. In addition, even if support were added for all systems and a common ontology emerged, a cognitive agent will still have its own interpretation of the world on which it bases its trust evaluations: thus trust must always be considered in the light of *why* agents trust each other.

Abdul-Rahman and Hailes' reputation model [6] approaches the problem of alignment from another direction, by defining the trust evaluations based on the actual communications. The interpretation of gossip is based on previous interactions with the same sender. The problem with this, however, is that it is incomplete: firstly it assumes all other agents in the system use the same model, which in a heterogeneous environment cannot be assumed. Secondly, it uses a heuristic based on prior experiences, to "bias" received messages. This bias is an average of all previous experiences. They do not differentiate between recommendations about different agents, which are based on different types of interactions.

Semantic alignment based on interactions has been studied in [7]. This approach to semantic alignment is based on the general framework of Channel Theory [8, 9]. We use this same mathematical theory as a framework for aligning trust.

## 2 The Algorithm

Channel Theory is a qualitative theory modeling the flow of information in distributed systems. From our point of view we can use this to describe a channel in which information about trust can be transferred from one agent to another. This is described in detail in [10]. The intuition is that both agents can relate each others' subjective trust evaluations to the objective descriptions of interactions, communicating about them using the languages $\mathcal{L}_{Trust}$ and $\mathcal{L}_{Domain}$, respectively. By doing so they are able to find the underlying meaning of the trust evaluations. The computational model based on this approach is described in [11] and we will summarize it here before explaining our implementation.

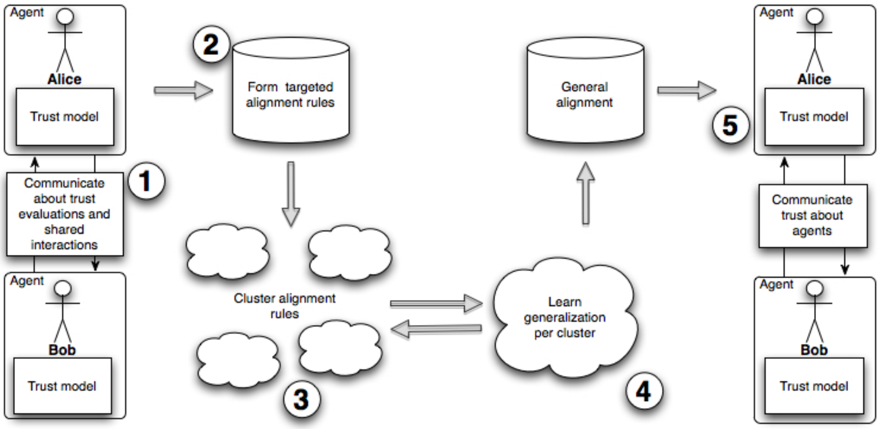

Agent
Alice
Trust model
2
Form targeted alignment rules
General alignment
Agent
Alice
Trust model
5
Communicate about trust evaluations and shared interactions
1
Communicate trust about agents
Agent
Bob
Trust model
Cluster alignment rules
Learn generalization per cluster
3
4
Agent
Bob
Trust model

**Fig. 1.** Schematic diagram of the steps in the alignment process

In Figure 1 we give a graphical overview of the algorithm. First, at ① the agents have to communicate their trust evaluations to each other in the form of `gossip` messages. For each message, the receiving agent computes its own trust evaluation, leading to a set of Specific Rules for Alignment (SRAs) at ②, each of the form $\alpha_i[T_j] \leftarrow \beta_i[T_j], \psi_i$, which would be the $i$th SRA about the target agent $T_j$. The heads of the rules $\alpha$ are the own trust evaluations, while $\beta$ in the bodies are the other agent's. $\psi$ describes the set of interactions which support both evaluations. The agent then has to learn the model underlying these SRAs. This is done, in repeated steps of clustering ③ and inductive learning ④. The output will be a set of General Rules for Alignment (GRAs), which is the generalization of the SRAs we give as input. These can then be used to interpret future messages ⑤. We describe the method used in Algorithm 1.

We use three important procedures, the clustering algorithm in line 5 and the two generalization algorithms we use on the clusters in lines 9 and 11.

- *Clustering* is used to group those SRAs where the *receiving* agent's trust evaluations are "near each other", because we want to learn generalizations that will predict that agent's trust evaluations, based on the gossip sent. That means we cluster based on the heads of the SRAs and we have an additional requirement for $\mathcal{L}_{Trust}$: there must be a distance measure defined on it. An

**Algorithm 1**: Generalize SRAs

```
Input: $\mathcal{R}$, the set of SRAs to be generalized
Input: $D(x, y)$, a distance measure on $\mathcal{L}_{Trust}$
Input: S, a set of increasing distances for clustering
GRAs := $\emptyset$
Clusters := $\{\{r\}|r \in \mathcal{R}\}$
Covered := $\emptyset$
foreach Stop_criteria s in S do
    Clusters := agglomerative_clustering(Clusters, s, D)
    if —Clusters— = 1 then
        break
    foreach $C \in$ Clusters do
        H := generalize_head(C, $\mathcal{R}$\C)
        if $H \neq$ null then
            G := generalize_body(C, $\mathcal{R}$\C)
            if $G \neq$ null then
                GRAs := GRAs $\cup$ $\{\langle$H $\leftarrow$ G, s$\rangle\}$
                Covered := Covered $\cup$ C
    if Covered = $\mathcal{R}$ then
        break
Output: GRAs
```

example of such a distance measure is described in Section 3.3. The clustering fulfills another role in the algorithm: it allows for the incremental learning of the alignment. This allows us to stop the algorithm when a suitable alignment is found. To this purpose we use a list of stop criteria S. This is a list of maximum distances for the clustering algorithm. The algorithm goes through this list from smallest distance to greatest and continues merging clusters until all the clusters are at a distance greater than the current stop criterion being evaluated. The resulting set of clusters will serve as input for the learning algorithms.

- *Generalizing* the SRAs is the main part of the algorithm. For each stop criterion s we will have a set of clusters of SRAs and for each of these clusters we shall attempt to generalize a set of GRAs covering it. Our first task is to learn a generalization of the heads of the SRAs. By definition, all the $\alpha_i$ within a cluster are within distance s of each other and we want to find some defining quality of these $\alpha_i$ which we can use in our final ruleset. We want to learn the generalization $\alpha^*$ which $\theta$-subsumes [12] all $\alpha_i$. Afterwards, when we generalize the body, we are learning the conditions for which the receiving agent should have trust evaluation $\alpha^*$. For both these tasks we can use an inductive learning algorithm, however the specific type of learning differs. For generalizing the heads of the SRAs we use an algorithm specialized in the "learn from example" setting [12], whereas for generalizing the bodies, we have richer information available and by using the "learn from interpretation" setting [12] we can use heuristics which take advantage of this, resulting in a faster algorithm for similar problems.

If we can find a generalization for the body it means we have a GRA which covers all of the targeted rules in the cluster. We stop the algorithm when all SRAs are covered, or when the remaining clusters are further apart than the largest stop criterion. When the algorithm ends we have a list of GRAs. This list can be used to translate messages from the other agent. Because each GRA is stored with the stop criteria which allowed it to be generated, we have an internal distance of the cluster it covers. We use this as the measure of accuracy of the alignment. We can use this, together with the actual aligned message, in the trust model.

## 3 Implementation and results

Now that we have described the problem and given an explanation of the system, we will give a brief description of the tools used to implement it. The implementation must:

- define a language for $\mathcal{L}_{Trust}$ and $\mathcal{L}_{Domain}$.
- implement the incremental clustering algorithm.
- use a "learn from example" ILP algorithm on the heads of SRAs.
- use a "learn from interpretation" ILP algorithm on the bodies of SRAs.

The implementation is predominantly in Java, which allows for flexibility in the tools used. For logical reasoning we use SWI-Prolog, which provides a JNI interface, so it can be accessed from Java.

### 3.1 $\mathcal{L}_{Trust}$ and $\mathcal{L}_{Domain}$ in OWL

To be able to gossip, the agents need two separate languages. One for the trust evaluations and one domain dependent language to talk about interactions. Often agents will be developed for domains where there already is a fairly extensive domain language available, so it makes sense to adopt this language as our $\mathcal{L}_{Domain}$. If there is no such global language, then agents will need to align their domain languages first. This is a separate problem and its solution is outside the scope of this paper. We assume there is a shared $\mathcal{L}_{Domain}$ language and we follow the W3C recommendation for its specification: we use the Web Ontology Language (OWL) [13]. Statements in $\mathcal{L}_{Trust}$ and $\mathcal{L}_{Domain}$ are expressed in a subset of OWL-DL in which we will not allow quantification at this point, because our learning algorithms are unable to deal with quantified variables in the examples. This is not a very big restriction, because we are always gossiping about one specific target agent, based on certain specific interactions and we can give the specific instance, without using quantification.

A remark about the semantics of $\mathcal{L}_{Trust}$: while we use OWL-DL to specify the language, we only fix the semantics of the connectives. The meaning of the predicates themselves is precisely what we want to align.

### 3.2 Clustering and Learning

As described in Section 2, agglomerative clustering methods best fit our needs. In this family, complete-link clustering [14] creates balanced clusters without requiring the computation of some form of centroid or medioid of the cluster. The calculation of such a centroid in our example is computationally intensive, as it is equivalent to generalizing the head of the SRAs. We will still need to compute this generalization, but not when using it as a distance measure, but only once the cluster is finalized. A drawback of complete-link clustering is that it deals badly with outliers. However, we are clustering on the agent's *own* trust evaluations. If there are outliers, they will not be in these evaluations, but rather the SRA itself will be an outlier. We will need to deal with the outliers in the learning of the body, but we should not encounter them when clustering. Our implementation is based on the algorithm described in [15], sufficiently optimized, in Java. The distance measure on the clusters depends on the distance measure on the elements, which are the statements in $\mathcal{L}_{Trust}$. Our implementation runs in $O(n^2)$ time, where $n$ is the number of SRAs formed from communication.

**Aleph and Tilde** For the generalization of the heads of all SRAs in the cluster we use Aleph [16], because it functions well at the "learn from example" setting. For learning the generalization of the bodies we use TILDE [17], which was designed for "learning from interpretation". Because the clusters aren't known beforehand and both these algorithms run with a specific format of input files, we generate the files for each cluster at runtime. Then we call the algorithm on the newly created files. The output is then read and reformatted so the head and body together form a complete Prolog clause. If both steps succeed, they result in a GRA, which covers the SRAs in the cluster. The algorithm terminates when all SRAs are covered in this manner.

### 3.3 Example environment

So far we have explained the implementation of the algorithm. To test it we designed an experimentation environment with a specific trust and domain language, a way of generating interactions and agents with different trust models. We base this environment on the following example scenario: Alice is looking for a keynote speaker for a conference and wants advice from Bob on who is trustworthy. This forms the basis of our example, with Bob and Alice trusting different agents based on different criteria. We form a random graph of interactions, which may range from an extremely sparse to a complete network, depending on the chance agents interact. The number of interactions between agents is customizable in a configuration file by varying the number of agents in the system and the chance agents interact. So far we haven't added support for varying the topology of the network, because testing the alignment depends mainly on the quantity of interactions, rather than the shape of the network. The ontology for $\mathcal{L}_{Domain}$ is defined in Figure 2. It is kept deliberatively small and describes only objective properties of the interactions.

Based on these interactions, the agents compute trust evaluations, which is communicated using $\mathcal{L}_{Trust}$. In this example $\mathcal{L}_{Trust}$ only has one predicate: image(Target, Value)[1], with $Value \in [1, 10]$. The distance between two elements image($T_1$, $V_1$) and image($T_2$, $V_2$) is defined as $\frac{|V_1 - V_2|}{10}$. This is the normalized distance between the two values. We purposefully disregard which agent is evaluated, because we want the clusters to generalize over all similar trust evaluations, ignoring which agent is being evaluated.

**Trust Models** We specify the trust models in our environment using Prolog programs. The relevant parts of the two models we use in our example are outlined in pseudocode in Table 1. The interactions are described in $\mathcal{L}_{Domain}$, which allows for 3 types of interactions: writing an article, a lecture and a personal interaction. These are observed and the participants in the interactions

[1] image is the trust evaluation of the target that the agent believes in

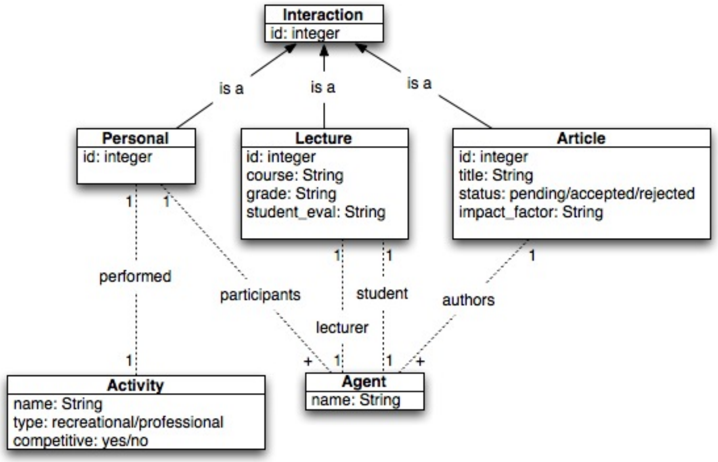


**Fig. 2.** The ontology for $\mathcal{L}_{Domain}$, in a UML-like representation

are evaluated in the trust models. We see that there are correspondences between the trust models, such as an interaction resulting in a high impact article, supports a high value image in both models. However we also see that Alice's model is less general than Bob's in this aspect: any article with an impact factor higher than 6 results in Alice evaluating the authors thereof with an image with value 10 (line A1). Bob's model splits this into articles with an impact factor higher than 8 (B1) and an impact factor between 6 and 8 (B4). Vice versa, Bob's trust model is quite general regarding lecture interactions where the target to be evaluated is the lecturer (B9). Any such interaction supports a trust evaluation with an image of 5. Alice's trust model considers such interactions in a more fine-grained manner, using various different rules (A2, A3, etc.). Finally we see a large difference between the trust models in that Bob's trust model has quite a few trust evaluations supported by lecture interactions where the target is a student (B2, B3, etc.). These simply do not correspond to any trust evaluation in Alice's model. These characteristics make these trust models useful for analyzing the running of the algorithm, because they cause the problems we typically expect to encounter when aligning.

Alice and Bob's knowledge bases contain the shared interactions between them. Each agent can therefore, based on these interactions, compute all possible trust evaluations for all agents in the system. In our experiments we consider Alice aligning with Bob, so we have Bob compute his trust evaluations and generate all the messages he can. These are sent to Alice, who uses them as input for the alignment algorithm. The reverse problem of Bob aligning with Alice is analogous.

### 3.4 Experiments and analysis

We run the above environment with stop criteria 0, 0.1, 0.5, 0.8 and 0.9. These give the increasing internal distance of the clusters we will attempt to learn generalizations for. The interactions are a randomly generated set, based on

**Alice's trust model**

```
A1  image(T, 10) ⟵ article(I), author(I, T) impact_factor(I, Imp), Imp > 6.
A2  image(T, 8) ⟵ lectured(I), lecturer(I, T), student_eval(I, excellent).
A3  image(T, 8) ⟵ lectured(I), lecturer(I, T), grade(I, G), G > 6.
A4  image(T, 7) ⟵ personal(I), participant(I, T), activity_type(I, recreational).
A5  image(T, 7) ⟵ lectured(I), lecturer(I, T), student_eval(I, good).
A6  image(T, 7) ⟵ article(I), authors(I, T), impact_factor(I, Imp), Imp > 4, Imp < 7.
A7  image(T, 6) ⟵ lectured(I), lecturer(I, T), student_eval(I medium).
A8  image(T, 6) ⟵ article(I), author(I, T), evaluated(I, accepted).
A9  image(T, 5) ⟵ article(I), author(I, T), evaluated(I, rejected).
A10 image(T, 4) ⟵ personal(I), participant(I, T), activity_class(I, competitive).
A11 image(T, 3) ⟵ lectured(I), lecturer(I, T), student_eval(I, bad).
A12 image(T, 1) ⟵ lectured(I), lecturer(I, T), student_eval(I, awful), !.
```

**Bob's trust model**

```
B1  image(T, 10) ⟵ article(I), author(I, T), impact_factor(I, Imp), Imp > 8.
B2  image(T, 10) ⟵ lectured(I), studied(I, T), student_eval(I, excellent).
B3  image(T, 10) ⟵ lectured(I), studied(I, T), grade(I, G), G > 8.
B4  image(T, 8) ⟵ article(I), author(I, T), impact_factor(I, Imp), Imp > 6.
B5  image(T, 8) ⟵ personal(I), participant(I, T), activity_type(I, recreational),
                   activity_class(I, cooperative).
B6  image(T, 7) ⟵ lectured(I), studied(I, T), student_eval(I, good).
B7  image(T, 7) ⟵ personal(I), participant(I, T), activity_class(I, cooperative).
B8  image(T, 7) ⟵ personal(I), participant(I, T), activity_type(I, recreational).
B9  image(T, 5) ⟵ lectured(I), lecturer(I, T).
B10 image(T, 5) ⟵ article(I), author(I, T), evaluated(I, accepted).
B11 image(T, 4) ⟵ lectured(I), studied(I, T), grade(I, G), G < 6.
B12 image(T, 4) ⟵ lectured(I), studied(I, T), student_eval(I, medium).
B13 image(T, 4) ⟵ lectured(I), studied(I, T), student_eval(I, bad).
B14 image(T, 2) ⟵ personal(I), participant(I, T), activity_type(I, professional),
                   activity_class(I, competitive).
B15 image(T, 2) ⟵ lectured(I), studied(I, T), student_eval(I, awful).
B16 image(T, 1) ⟵ article(I), author(I, T), evaluated(I, rejected).
```

**Table 1.** Two sample trust models in Prolog-like syntax

different configuration settings. We use 60% of these interactions as our training set and 40% as the control set. We run thirty trials at each configuration and Table 2 shows a summary of the results, where each row is the average over the trials run. The first two columns describe the configuration and the third the number of interactions the configuration results in. The other columns we describe below.

| Num. agents | Interact chance | Interact number | Coverage training | Coverage control | Accuracy |
|---|---|---|---|---|---|
| 10 | 10% | 14 | 75% | 45% | 0.99 |
| 10 | 30% | 40 | 89% | 70% | 0.97 |
| 10 | 70% | 95 | 99% | 95% | 0.92 |
| 25 | 1% | 7 | 69% | 41% | 0.96 |
| 25 | 5% | 44 | 78% | 66% | 0.95 |
| 25 | 10% | 96 | 96% | 93% | 0.95 |
| 25 | 30% | 275 | 94% | 94% | 0.92 |
| 25 | 50% | 451 | 98% | 98% | 0.87 |
| 50 | 2% | 74 | 76% | 65% | 0.96 |
| 50 | 10% | 373 | 97% | 97% | 0.94 |
| 50 | 30% | 1096 | 100% | 100% | 0.99 |

**Table 2.** Summary of results

**Coverage of the Alignment** Our algorithm always results in a "catch all" GRA at the highest stop criterion of 0.9. All SRAs are covered by this rule, however it adds no information. We discuss this further in Section 3.4. The coverage in Table 2 is calculated as the percentage of SRAs that are covered by any GRA other than this "catch all" GRA.

We see that even in a fairly small shared network, such as the one with 10 agents and a 30% interaction chance the alignment achieves a fairly high coverage of 70%. This was based on an average of 40 interactions which led to an average of 48 SRAs for aligning. The networks smaller than that do not allow for good alignment. With 10 agents and 10% chance of interacting the coverage drops below 50% with quite a few instances of runs with 0% coverage of the training set. This is based, on average, on 14 interactions. The trials with 25 agents and 1% chance of interaction suffer even more under the network size: with only around 7 interactions to base the alignment on, 5 of the 30 trials did not complete. Of course, by using all interactions, and not just 60%, an agent can improve its learning capacity in small networks, but in this example around 20 interactions is the minimum for a reasonable alignment. This improves fairly rapidly if agents have more shared interactions, because, as is to be expected, the quality of the alignment is mainly dependent on the number of interactions. We see for instance that the trials with 25 agents and a 5% chance of any two agents interacting on average finds an alignment that covers 66%, a vast improvement over the network with 1% interaction chance. Number of interactions is not the only factor influencing the coverage of the alignment. Sometimes the training set does not allow for a good alignment and the training set has bad coverage. Luckily, there is a correlation between the coverage of the training set and the control set: the Pearson's correlation coefficient is 0.51[2] and if we do not take the data from the two smallest sets into account, which do not result in decent alignments, the correlation is stronger, with a coefficient of 0.72[3]. In Figure 3(a) we have graphed the coverage of the training and control set for trials with a small sized network: with 50 agents and a 2% chance of interacting, displaying this correlation. This correlation is useful, because an agent can know the coverage of its training set and thus if the training set has low coverage it can expect bad results when applying the alignment.

The knowledge about the training set is useful. If the training set is small or has low coverage the agent knows that its alignment probably has a low coverage for any future messages from the other agent too. Furthermore, it can attempt to improve the alignment by obtaining information about more interactions and the other agent's trust evaluations based on them. We show the improvement an agent can achieve by ignoring badly covered training sets. In Table 3 we give the average coverage of the control sets if we leave out all alignments that cover less than 70% of the training set, showing a marked improvement in the trials run in smaller networks. In medium networks with over 100 interactions we can achieve a similar improvement by leaving out all alignments with less than 100% coverage of the training set. In the largest networks, with over 500 interactions the coverage of both sets is so near perfect that there is no significant improvement.

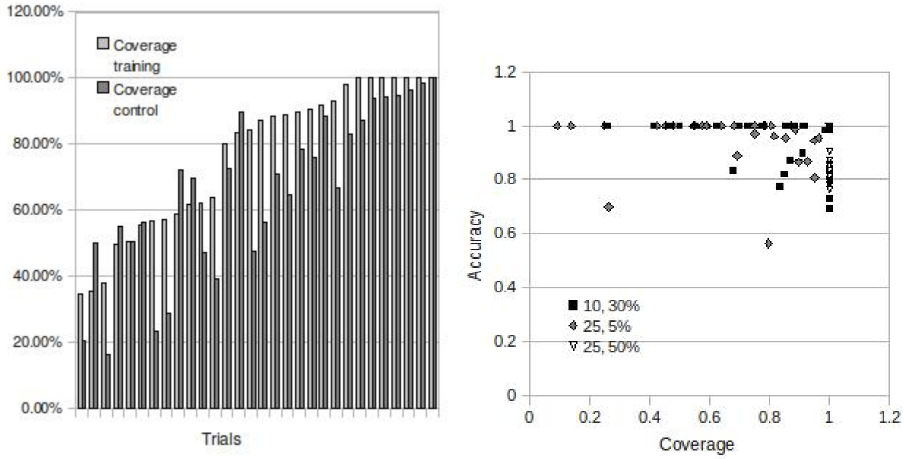
| Experiment | 10, 10% | 10, 30% | 10, 70% | 25, 1% | 25, 5% |
|---|---|---|---|---|---|
| Coverage | 59% | 75% | 96% | 49% | 77% |
| Number ignored | 9 | 5 | 2 | 9 | 12 |

| Experiment | 25, 10% | 25, 30% | 25, 50% | 50, 2% | 50, 10% |
|---|---|---|---|---|---|
| Coverage | 97% | 98% | 100% | 81% | 100% |
| Number ignored | 3 | 2 | 6 | 12 | 5 |

**Table 3.** Corrected coverage of control set, by removing the trials with bad coverage of the training set

[2] With 95% confidence we know the correlation coefficient is between 0.44 and 0.59

[3] With 95% confidence we know the correlation coefficient is between 0.66 and 0.77



(a) Coverage of training and control sets

(b) Coverage vs. accuracy

**Fig. 3.** Graphs showing various aspects of the coverage

**Accuracy of the Alignment** As mentioned in Section 2, the accuracy of a GRA is directly proportional to the maximum distance of the SRAs it covers, in fact, the accuracy is $1 - \mathtt{s}$, with $\mathtt{s}$ the stop criterion used to generate the cluster of SRAs. We define the accuracy of the alignment as the average of the accuracy of the GRAs used to translate each message in the control set. Thus any uncovered message is not taken into account in the accuracy calculation. In most cases we find an extremely accurate alignment. Often, the highest accuracy is when coverage is low, while the alignments with higher coverage have a lower overall accuracy. This can be explained by looking at the way the learning algorithm works: when we decrease accuracy by moving to a large clustering distance, any GRAs learned will have a higher coverage. Most alignments in the small and medium networks with a high coverage have one or two GRAs with lower accuracy. We have plotted the relation between accuracy and coverage for a few trials in Figure 3(b), where we see that lower accuracy occurs more often at high coverage.

Additionally, for our example, we can look at the trust models to see that an accuracy of 100% is a theoretical impossibility. In some situations, especially those concerning lecture interactions, Alice's trust model is far preciser than Bob's, leading to messages from Bob necessarily being aligned with a lower accuracy. We expect this to be a general feature of two different trust models.

If we take coverage into consideration, the average accuracy drops to around 40% for the smallest data sets. Larger networks aren't influenced as much by this correction, because coverage tends to be higher and accuracy already lower. The large networks have similar accuracy before and after the correction: around 90%.

**Unaligned messages** As described in Section 2, there is another measure of the coverage of the alignment, which we have so far ignored: the messages Bob sends which are based on interactions which support no corresponding trust evaluation for Alice, for instance in our example Alice cannot use any trust evaluation Bob bases on `lecture` interactions in which the student is evaluated. These cases are not taken into account in the data displayed above, because there are two ways of considering them. One possibility is to consider any such message as successfully aligned: the agent knows these messages do not result in a corresponding trust evaluation. We could therefore use the learner to find a generalization of these messages in the same manner as we learn the body of messages that do result in an alignment. In this example these messages can easily be generalized: they are either the `lecture` interactions described above, or `personal` interactions with a cooperative and professional activity. Learning this generalization results in a 100% coverage for both training and control groups and thus Alice can know in the future when a message does not correspond to an own trust evaluation. The other possibility is for an agent to consider such messages unaligned. This results in an overall lower coverage of both the training and control sets. In this example 21% of the messages Bob sends do not result in a trust evaluation for Alice.

We prefer the first way of considering these messages, because knowing in which situations a trust evaluation of the other agent does not correspond to any trust evaluation in the own agent is valuable information in and of itself and can even be considered as a successful alignment: knowing *when* there is no information is also information!

## 4 Conclusion

The problem we address in this paper is that of aligning trust models. We describe the implementation of an alignment algorithm and have performed a preliminary set of experiments with it. The experiments show that even at low numbers of interactions the coverage and accuracy of the alignment is quite high, although for a reliable 100% alignment it requires large amounts of interactions. Luckily, just by analyzing the coverage of the training set, we can estimate whether the alignment will be effective. This gives the agent an extra tool in the application of the alignment. Furthermore we see that the average accuracy is high for aligned messages.

The trust models we used are simple and in the future we will test the algorithm with actual trust models, used in the community. We will also test it in a scenario with real data, rather than randomly generated interactions. This approach will necessitate more robust accuracy checks and outlier detection, because in these experiments we did not deal with noise or inaccurate trust evaluations. The ILP algorithms used can be configured to deal with noisy data, however we did not do this in these experiments: we only use results from Tilde with a 100% coverage of the examples, by allowing results with less coverage, we would need to combine the accuracy results from Tilde with our own accuracy calculations. Because there are various ways of doing this and it was unnecessary for the example scenario we have not considered this.

## References

1. Sabater-Mir, J., Paolucci, M., Conte, R.: Repage: REPutation and imAGE among limited autonomous partners. JASSS - Journal of Artificial Societies and Social Simulation **9**(2) (2006)
2. Hübner, J.F., Lorini, E., Herzig, A., Vercouter, L.: From cognitive trust theories to computational trust. In: Proc. of the Twelfth Workshop "Trust in Agent Societies" at AAMAS '09, Budapest, Hungary (2009) 55–67
3. Pinyol, I., Sabater-Mir, J.: Arguing about reputation. the lrep language. In Artikis, A., O'Hare, G., Stathis, K., Vouros, G., eds.: Engineering Societies in the Agents World VIII: 8th International Workshop, ESAW 2007. Volume 4995 of LNAI., Springer Verlag (2007) 284–299
4. Casare, S., Sichman, J.: Towards a functional ontology of reputation. In: AAMAS '05: Proc. of the fourth international joint conference on Autonomous Agents and Multiagent Systems, Utrecht, The Netherlands, ACM (2005) 505–511
5. Nardin, L.G., Brandão, A.A.F., Muller, G., Sichman, J.S.: Effects of expressiveness and heterogeneity of reputation models in the art-testbed: Some preliminar experiments using the soari architecture. In: Proc. of the Twelfth Workshop "Trust in Agent Societies" at AAMAS '09, Budapest, Hungary (2009)
6. Abdul-Rahman, A., Hailes, S.: Supporting trust in virtual communities. Proceedings of the 33rd Hawaii International Conference on System Sciences **6** (2000) 4–7
7. Atencia, M., Schorlemmer, M.: I-SSA: Interaction-Situated Semantic Alignment. In: OTM 2008, Part I. Volume 5331 of LNCS., Springer (2008) 445–455
8. Barwise, J., Seligman, J.: Information Flow: The Logic of Distributed Systems. Cambridge University Press (1997)
9. Schorlemmer, M., Kalfoglou, Y., Atencia, M.: A formal foundation for ontology-alignment interaction models. International Journal on Semantic Web and Information Systems **3**(2) (2007) 50–68
10. Koster, A., Sabater-Mir, J., Schorlemmer, M.: A formalization of trust alignment. In Sandri, S., Sànchez-Marré, M., Cortes, U., eds.: AI Research and Development. Proc. of the Twelfth International Congress of the Catalan Association of Artificial Intelligence (CCIA 2009). Volume 202 of Frontiers in Aritficial Intelligence and Applications., Cardona, Spain, IOS Press (2009) 169–178
11. Koster, A., Sabater-Mir, J., Schorlemmer, M.: Inductively generated trust alignments based on shared interactions (extended abstract). In: Ninth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2010), Toronto, Canada, IFAAMAS (In Press)
12. De Raedt, L.: Logical and Relational Learning. Springer Verlag (2008)
13. McGuinness, D.L., van Harmelen, F.: Owl web ontology language overview. http://www.w3.org/TR/owl-features/, retrieved July 27, 2009
14. Defays, D.: An efficient algorithm for a complete link method. The Computer Journal **20**(4) (1977) 364–366
15. Johnson, S.C.: Hierarchical clustering schemes. Psychometrika **32**(3) (September 1967) 241–254
16. Srinivasan, A.: The aleph manual. http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/, retrieved 9/2/2009 (June 2004)
17. Blockeel, H., Dehaspe, L., Demoen, B., Janssens, G., Ramon, J., Vandecasteele, H.: Improving the efficiency of inductive logic programming through the use of query packs. Journal of Artificial Intelligence Research **16** (2002) 135–166

# Why does trust need aligning?

Andrew Koster

IIIA, Artificial Intelligence Research Institute
CSIC, Spanish National Research Council
Bellaterra, Spain
andrew@iiia.csic.es

**Abstract.** In this position paper we explain why the alignment of trust for computational agents is a problem which requires closer consideration than it has previously been given. We give a review of related work from various fields of research and propose a general framework in which a solution for the alignment of trust should be found.

## 1 Introduction

*de gustibus non est disputandum*
— Latin maxim

One of the main problems in open multi-agent systems is how the heterogeneous agents can interact with one another. Usually an ontology is given by the system designers for the agents to communicate. If there is no single ontology, but each agent uses its own, there is a large amount of work done to enable the alignment of these ontologies [1]. An ontology can be used to describe the environment and negotiate in objective terms. However, if the objects to be discussed are subject to semiotic heterogeneity [2], ontological solutions do not suffice. Semiotic, or pragmatic, heterogeneity is the problem encountered when an ontology can be interpreted in different manners. The agents agree on the syntax of the ontology and the content the ontology describes, yet still do not coincide on the *meaning* of the concepts. This is the problem we encounter when talking about taste... and also when talking about trust. We do not even have a non-ambiguous definition of what trust itself means. There are various philosophical, sociological, cognitive and economic theories of what trust means to humans [3–7] and different computational models based on these [8, 9]. Agents within an open MAS using such diverse models of trust and reputation will therefore run into communication problems if they wish to exchange information about these concepts. In [10] a strong case is made for why agents need to communicate their trust information and the manner of reputation formation is solely based on communication. As an example we can look at eBay's system [11]: their website does not include any tools for modeling trust, just an interface for communicating evaluations. The other users use this information to form their own evaluations. This simple system is one of the pillars for the website's tremendous success [12]. One of the reasons it works well is because it allows users to write a comment: not only do they give a score to the trade partner, but a short explanation *why* they give that score.

When we state opinions, we almost always give the reason for having this opinion. The same holds true for opinions about trust. Consider the phrase "I wouldn't trust Alice". If uttered sincerely, this is uttered in accordance with Grice's maxim of quality [13]: the speaker believes it to be true and the speaker can justify it based on some evidence. Yet in reality we are not so easily satisfied and we have the urge to ask for a justification from the speaker. The recognition that a justification is required, is already present in toddlers [14] and when given, such as in an utterance "I wouldn't trust Alice, because the car she sold me was a lemon", it is far more likely to be accepted by a listener. Of course, we are assuming that a "lemon" is the same to both the speaker and the listener. If not, this is an equally unsatisfactory utterance and a further explanation of what the speaker considers to be a lemon is required. We therefore continue asking for justifications of opinions until we are satisfied we understand what the speaker *means* (in [14] this is referred to as the justifications being grounded) and there is no further semiotic heterogeneity. Whether or not the listener agrees with the speaker is another issue entirely. It may very well be the case, that the listener is perfectly happy to trust Alice in her role as a car saleswoman, despite the speaker's utterance. It might even be the case that the listener disagrees with the reasoning of the speaker: Alice selling him a lemon is not a good reason to mistrust her. Perhaps even to the contrary: the listener rather dislikes the speaker and Alice selling him a lemon is all the more reason to esteem Alice. However, whatever complex reasoning lies behind the conversation partners' opinions, for them to communicate effectively they need to justify their opinions. There is evidence that this necessity of justifying a statement appears when we first begin to realize that the communication partner has mental states which may differ from our own [15]. This same difference in mental states should be assumed in autonomous computational agents [16] acting in an open MAS. All agents participate in the MAS to fulfill their own goals, based on their own beliefs, therefore we should expect communication of opinions, including opinions of trust, to be accompanied by some justification.

In this paper we discuss the problem further as well as giving a brief description of a proposed method for aligning trust. In the next section we discuss methods related to alignment of trust, such as argumentation, ontology alignment and dealing with uncertainty. In Section 3 we further expand on the problem and propose an area in which the solution could be found. We recap the main points and discuss them at the end of the paper.

## 2 Related work and similar problems

As mentioned in the previous section, trust alignment falls within the general area of semiotic heterogeneity, which is recognized as a sub-area of general semantic heterogeneity. While most work done on semantic heterogeneity focuses on the more traditional forms of ontology alignment, there is some work done on the problem. The field of semiotics [17] focuses on how humans interpret signs and how meanings are formed. Some work has been done on applying semiotics to AI, most notably semiotic dynamics [18] studies how semiotic systems come into being, both in human societies and in artificial societies. However, so far its application to alignment problems has been slim. Most work concentrates on evolving a semiotic system together, thus reaching a consensus of meaning. However, this is not possible with all concepts. While allowing agents to evolve a language to talk about external objects is possible, when they talk about their own mental states we specifically do not want a consensus. Each agent has their own way of interpreting the world and it is this interpretation they must try to communicate.

Communicating trust evaluations is exactly such an issue and some work has been done in trust modeling for computational systems to address it. Most trust and reputation models take both the agent's experiences and other agents' communications into account, however Abdul-Rahman & Hailes' model [19] recognizes differences in mental states to a certain extent: the interpretation of communicated trust is based on previous interactions with the same sender. This allows their model to use a heuristic to "bias" received messages depending on how far apart received trust evaluations from the same sender in the past have been from the agent's own trust evaluations. We agree that this setup is a good approach to the problem. By learning quantification of the dissimilarity between the agent's own trust evaluations and the communicated trust evaluations received from the other agent, a form of an alignment is made. However, their model misses out on some important points:

- Their alignment system only works with other agents using the same trust model. However, in an open MAS this cannot be assumed. This might be solvable using regular ontology alignment before the "biasing" of the message. Various trust ontologies have been proposed [20, 21] as well as ontology mapping service for them [22].
- More serious is the fact that they do not take the context of the trust evaluation into account. The reason trust is subjective is because each agent has its own goals and observations of the environment in which it evaluates trust. This may cause the bias to vary between trust evaluations. Their model simply averages the bias into one general numerical bias. This is a simplification we feel cannot be made.

Dealing with the context in which a trust evaluation is made complicates matters, because the bias has to be made conditional upon this context, which needs to be discovered in its own right. The approach we propose in Section 3 is inspired both by this model and the insight that context plays an important role.

Insofar as we know, this is currently the only model attempting any form of alignment of incoming trust information. There are models based on cognitive principles [23, 24] which offer the capability to do similar things, however the main focus of these models is on the cognitive modeling of trust and they leave the alignment of incoming messages as an open problem.

Another approach is to not use other agents' trust evaluations at all and instead communicate just the information about interactions. This is done in [25]. While this avoids any subjective terms in the communication, it does not work well if the information is asymmetric between the agents or there are privacy issues in communicating. For communication of just domain information to be effective, the agents have to communicate everything about an interaction, allowing the other agent to compute its own trust evaluation. A separate alignment process allows agents more freedom in communication: in this process agents only talk about interactions they have both observed. The agents both have different information available to them, which allows them to compute their own trust evaluations. The agents can then choose which properties of the interactions to communicate, in accordance with any restrictions they have. If similar situations consistently lead to similar communicated properties, then even if the other agent does not have knowledge of the interaction, it can approximate its trust evaluation through the similarity of previous situations.

### 2.1 Is argumentation about trust the same as aligning?

Some work has been done focusing on ways to build argumentations for trust [26, 27]. The reasons for argumentation are the same as the ones given above: agents need to explain their trust evaluations to each other [26] and to the user [27]. We agree that argumentation is an excellent domain to find a solution to the problem at hand: it gives a formal framework for building explanations. However, there are two important issues which are not answered by the work in argumentation so far. [26] describes an argumentation language in which agents can form justifications for their trust and communicate these to each other. Such a justification consists of a trust evaluation to be justified and a phrase in their justification language. However, their justification language consists of further predicates about trust, as well as agents' evaluations of interactions. This allows agents to build justifications for their ungrounded terms on further ungrounded terms. Somehow these terms need to be justified in grounded terms. As we saw in the introduction: the concept of a car being a "lemon" may be equally subjective as the trust based on that. Agents will need to justify *why* they evaluated the car as a lemon. This process should be repeated until the terms of the conversation are only terms in a shared, objective language describing the domain. The other problem is that there is no clear description of what agents should *do* with these justifications when they receive them. [26] says these can be incorporated into the trust model, however there is no description of how this should happen. Such a method of incorporation into the trust model, together with communication in objective justifications about trust is an alignment. [27] offers a different view: the justifications are used specifically to communicate to the user why a trust evaluation is given. The justifications are therefore output of the trust model to the user and the language can be grounded in the user's own terms. Furthermore they use an "opponent modeler", which learns to distinguish different situations in which recommendations are to be trusted or not. This opponent modeler could very well be seen as a method for learning an alignment, however it only takes the agent's own past experiences into account. This means it will need a large set of interactions with the same "opponent" agent to learn an accurate model, as their experiments seem to corroborate. Due to such interactions being prior to their modeling, or alignment, the agent runs the risk of such interactions being harmful, either due to malintent or simple miscommunication. Rather than only using the interactions the two agents have had with each other, the agents could learn an alignment, based on all interactions both agents have information of, thus reducing the risk from many interactions with an unknown opponent. This alignment could be formed in a separate communication process. Another facet of the approach in [27], is that they seem to focus on detecting when an opponent is dishonest. While an important facet, it is not the only situation in which an alignment method would be useful, as we will describe in the next section.

### 2.2 Is taking uncertainty into account sufficient?

There has been quite a lot of work done on discovering dishonesty in communicating trust evaluations. The main point of such research is to find a way of detecting *when* a communicated trust evaluation is inaccurate. In [28] these methods are divided into endogenous and exogenous methods. The endogenous methods discover unfair evaluations through statistical analysis of all ratings. This presumes that the meaning of trust is the same to all agents. Statistical methods can detect which agents diverge from the norm, but in an open MAS this doesn't automatically imply these agents are frauds. They may have different trust models. Additionally, if enough different trust models are in use, the significance of these methods drops considerably. These methods are designed to work in environments where few agents "lie", which is taken to mean their opinion deviates from the average. Thus if many different models are used, this assumption will not hold. Another assumption underlying these models is that the communication acts about trust are either public, or passed through a central unit where such statistical measures can be computed. These methods are therefore not very well adapted to use in an open MAS. The exogenous methods are more diverse and are defined by their use of additional information to determine unfair evaluations. TRAVOS [29] and BRS [30] for instance predict the reliability of a trade partner by calculating the expected value, given a probability distribution which is tailored to fit past experiences with that partner. TRAVOS in specific takes the context of the past experiences used in this calculation into account, thus discerning between similar and dissimilar situations to assess the reliability more accurately. POYRAZ [31] was developed as a combination of endogenous and exogenous methods and expands on TRAVOS' method, by taking not only the own experiences, but combining this with publicly available information, such as reputation. [31] shows experiments in environments with liars, in which POYRAZ and TRAVOS show a significant improvement over similar models which do not take contextual information into account. This confirms our earlier assertion that it is important to distinguish the context in which a trust evaluation is communicated.

However, there is an important issue which is not considered in the models discussed above: the question of *why* the information is unreliable. In the theory and in the experiments all these models make the assumption that the reason a communicated evaluation is unreliable is because it is either incompatible with the own model, or the agent is lying. We argue that these are two very different cases. In the first situation the agent's evaluation is different because it is based on a different trust model. In the second it is because the agent has malicious intentions. Models for dealing with unreliable information, however, deal with both situations in the same manner: the information is discarded. This should not be necessary in the first case, if only the agents can align their notions, such communications can be translated and used as reliable information. Furthermore, because the models don't distinguish between the two situations this may have repercussions for the truthful, but badly aligned agents, if it is assumed they are lying: when this information is propagated it may influence their reputation. Thus the statistical analysis is very necessary to discover *when* it might be useful to align, but it doesn't replace alignment. That would discard useful information, as well as negatively impacting the information-giver's reputation.

## 3 How to align trust?

One recurring theme, both in the theoretical approaches and in the related work we have discussed is a clear division between the subjective trust evaluations and the objective context information on which they are based. In [27] the argumentation is based on how the opponent is modeled, using the experiences the agent has had with that opponent. Similar experiences are used in TRAVOS and POYRAZ to discover unreliability in the trust evaluations. This is unsurprising, because these experiences play a central role in trust. A trust model evaluates agents based on such experiences. It can take only its own experiences into account, or also experiences the agent has observed. Furthermore such observations may be communicated. To be able to communicate about trust evaluations, it is therefore essential to also allow communication of these experiences. [31] gives an example ontology allowing for this, however each domain may have its own unique ontology to describe the interactions. In general a MAS will provide such an ontology to agents participating. An alternative is that agents have their own personal domain ontologies. These can be aligned using general ontology alignment methods [1]. Due to these being grounded ontologies about the environment we do not run into the problem of aligning subjective opinions. Once a shared domain ontology has been established the agents can exchange trust evaluations and information about the interactions such evaluations are based on. The receiving agent can use this information to form the alignment. We will describe the requirements for these parts in more detail, but first give a brief overview of the mathematical framework, giving more rigour to this idea.

### 3.1 Theoretical Foundations

Channel Theory [32] has been proposed as a general framework for semantic alignment [33]. This theory is a qualitative theory modeling the flow of information in distributed systems. [34] shows how dynamic situated ontology alignment can be considered in this framework. While this is a very different problem from the one we are considering, the article shows how a channel theoretic framework can aid, not only from a theoretic point of view, but also in considering how an alignment is formed. We can describe a channel in which information about trust can be transferred from one agent to another. This framework is described in detail in [35] and we give a short summary here. The intuition is that both agents can relate each others' subjective trust evaluations to the objective descriptions of interactions. By doing so they are able to find the underlying meaning of the trust evaluations.

**Interaction-based alignment** As we have argued, agents' interactions in the environment form the basic building blocks for trust. Such interactions are observed by different agents and each agent has an internal representation of this interaction. We make no further assumptions about such representations. As argued in [31], each agent may focus on different aspects of the interaction. Additionally agents may not receive the same information about an interaction. We further suppose that each agent may have its own way of representing such information.

These observations then lead to trust evaluations of the various agents involved. Any trust model can therefore be described as a binary relation between an agent's observations and its trust evaluations. These trust evaluations can be represented in some language $\mathcal{L}_{Trust}$, which we assume can be represented by all agents in the system. The meaning of phrases in $\mathcal{L}_{Trust}$ to the different agents is what the alignment process should uncover.

We consider trust alignment as a case-by-case problem. There is no need to align with agents there is no communication with. Furthermore, we could use a statistical analysis such as the ones used in POYRAZ or TRAVOS to filter the cases in which alignment is useful. Only in those cases should agents align. To do this, the agents need to discuss the interactions they both have observed and we assume there is a shared language to discuss these interactions. We call this language $\mathcal{L}_{Domain}$ and emphasize that it is a shared language: both the syntax and the semantics are known by all agents in the system, as opposed to the semantics of $\mathcal{L}_{Trust}$, which is interpreted differently by the agents. Each agent can relate its own internal representation of interactions to phrases in $\mathcal{L}_{Domain}$. Because this is a language about the objective, grounded, properties of the interaction, not all observations of an interaction can be communicated, however it allows us to define exactly what it means for two agents to share an interaction. A set of interactions $I$ is shared by agents $A$ and $B$ if there is some $\varphi \in \mathcal{L}_{Domain}$ such that $\varphi$ is in both $A$ and $B$'s sets of observations of the set $I$, or, in other words, $\varphi$ is the information shared between the agents about $I$. The information in $\varphi$ could range from a detailed description of the interaction to only the very basic fact that the interaction took place. This depends on what both agents know about the interaction and what they are willing to share as well as what can be represented in $\mathcal{L}_{Domain}$.

For aligning, the agents should *only* use the trust evaluations based on interactions *both* agents observed. For this shared *core* of interactions it is known that while each agent may have different observations, these come from the same interactions. By communicating only about the trust evaluations these interactions support, the agents guarantee that they are sharing the "building blocks" of those trust evaluations and the *channel* [32] of information flow is established.

**Forming the alignment** Note that such trust interactions are not necessarily the same trust evaluations that either agent actually uses: when functioning normally, most trust models use *all* information, and thus interactions, available to them. This results in "believed" trust evaluations. For alignment purposes, however, there is no reason to limit the agents to just these "believed" trust evaluations. To expedite the alignment process it is useful to consider all trust evaluations that *could be* supported by the shared interactions, rather than only those which, in fact, are.

The basis of a trust alignment is a set of messages sent from one agent to another in the form $\langle\beta, \psi\rangle$, with: $\beta$ a trust evaluation of a specific target in $\mathcal{L}_{Trust}$ and $\psi$ pinpointing the specific shared interactions this evaluation is based on in $\mathcal{L}_{Domain}$. We now see the framework for the alignment process arise, because if one agent $B$ sends such a message, the receiving agent $A$ can compute its own trust evaluation $\alpha$, based on observations of the same interactions supporting $\psi$. A Specific Rule for Alignment is thus made and is the tuple $\langle\alpha, \beta, \psi\rangle$.

These SRAs must now be generalized to a predictive model, such that, for example, agent $A$ can know what trust evaluation $\alpha'$ it should associate with a certain $\beta' \in \mathcal{L}_{Trust}$, given a description $\psi'$ of some interactions, which it has not observed. It must be able to associate a trust evaluation with only the communicated information $\beta'$ and $\psi'$.

### 3.2 Finding predictive rules

Channel theory gives us a theoretically sound manner to define the building blocks of an alignment, but the actual process of finding a useful set of rules, which will allow future communications to be translated is not captured by channel theory. In fact, we propose there are various methods of alignment possible and choosing which one to use is dependent on both the domain and the agents aligning. We will discuss some of the options here.

**An inductive approach** We start with an approach using inductive learning to find a generalization. There are various different inductive algorithms, but they all use $\theta$-subsumption at some point or another [36]. If we consider our SRAs in a slightly different format: $\alpha \leftarrow \beta, \psi$ and restrict the predicates used to first order logic, it is easy to see how these rules constitute a logic program. By using ILP [36] we can find generalizations of these, as described in [37]. The induced rules form a different logic program, which translates the other agent's trust evaluations, given a context $\psi$. A similar approach to modeling other agents is taken in [27], where a fuzzy rule induction algorithm is used in their "opponent modeler". While the focus is different, because the resulting rules are used for argumentation, the basic principle is the same. However, there is another difference between the approaches: the former uses induced rules to directly translate trust evaluations whereas the latter uses such rules as an approximation of the opponent's trust model. It would require a further step to translate trust evaluations from the induced model into similar ones for the own model. This approach of modeling and then translating was taken in [38].

An ILP-based approach has one major downside, which is that ILP algorithms are very dependent on the language bias. If we do not have enough information to structure the search space quite rigorously the algorithm will either not find the correct generalization or take prohibitively long to search for it. By giving enough background information the search space can be made far more accessible, however this task needs to be done manually. Depending on the domain, the associated ontology and the agents aligning this may or may not be viable. A method of automatically generating the language bias would be through analysis of the communicated messages only and not taking the entire ontology into account. This requires the agents to only communicate the relevant properties of the interactions in $\mathcal{L}_{Domain}$ and a method for dynamically constructing the language bias for the ILP algorithm. Insofar as we know, no work has been done towards such an approach.

**A context-discovery approach** Instead of considering the SRAs as a logic program, we could see them as $\psi$ being an example of the context in which $\alpha$ and $\beta$ are aligned. Therefore finding generally predictive alignment rules equates to the problem of finding a good classification of the contexts. Some work has been done in automatic context recognition for trust [39, 40] and as mentioned in Section 2.2, there are various models to detect unreliable information based on the context. An intuitive continuation of such approaches would be to apply context recognition not only to trust evaluations but to the alignment of communication about the same. One issue with clustering- or classification-based approaches is to find an appropriate distance measure. In approaches so far, propositional or attribute-value logics have been used to describe the contexts. If the language used is more complicated, the distance measure becomes harder to define, or calculate. An example of a distance measure which could work for first order logics is [41], however in general this is an open issue.

**An argumentation-based approach** Another way of considering the SRA is as an argument. $\psi$ is a justification for the trust evaluations $\alpha$ and $\beta$ in both agents respectively. While this justification is given in a domain level language it is easy to see how this could be extended. The argumentation language proposed in [26] allows agents to communicate their justifications for trusting an agent, however it misses the link to such a grounded language. By extending this language the agents could each present their justifications for their trust evaluations. Such argumentations could either be used to negotiate about trust, or as cases in a Case-Based-Reasoning [42] algorithm. This could be used as a predictive method, by retrieving a comparable justification in the past and the corresponding trust evaluations.

## 4 Discussion

We have argued that trust alignment is a real problem if multiple trust models are to be effectively used in an open MAS. So far this issue has superficially been addressed by various techniques to deal with argumentation of trust or discovering unreliable communications about trust. While giving an indication of the area in which a solution must be found, they do not solve the issue themselves. We feel the general framework of alignment presented in Section 3.1 gives a good basis to work on the problem and present possible approaches to solve the alignment problem. Our own work so far has focused on inductively generating logical rulesets as an alignment, however as mentioned, this approach is very dependent on the available background information. Whether this approach is suitable or not therefore depends to a large extent on the type of trust models used and the available $\mathcal{L}_{Domain}$. In general we expect that the best approach will be largely context dependent. Some factors which may play a role in deciding which approach to use are:

- Number of agents. If there are many agents and a large number of interactions the learning-based approaches may be more suitable, while for domains with a small number of agents, argumentation may be better.
- Expressivity of $\mathcal{L}_{Domain}$. Because ILP-based solutions require a rich language, if this is not available, context-discovery may be a more suitable approach. Similarly argumentation requires an argumentation language.
- Complexity of trust models. If trust models used are very complex, machine-learning approaches may not be able to handle the task. Argumentation-based approaches may give better results.

We also do not claim to give a complete overview of methods. There may very well be other approaches we have not thought of, or combinations of the ones we have mentioned. We feel there is a lot of work to be done in this domain and we have only touched the tip of the iceberg.

Methods for discovering dishonesty should play an important role throughout the process. Not only do we feel the methods proposed could be used to find when agents could benefit from aligning, but there is a possibility for dishonesty during the alignment process. One advantage of aligning is that, because the alignment process relies on objective information, being dishonest is harder. Furthermore the results are less predictable, because the agents' trust evaluations are translated into each others', rather than being incorporated as "truthful" information. Lying during the alignment process is therefore harder, however we do not rule out the possibility and statistical methods to discover liars remain important.

## References

1. Euzenat, J., Shvaiko, P.: Ontology matching. Springer-Verlag, Heidelberg (DE) (2007)
2. Bouquet, P., Ehrig, M., Euzenat, J., Franconi, E., Hitzler, P., Krötzsch, M., Serafini, L., Giorgos, S., Sure, Y., Tessaris, S.: D2.2.1 specification of a common framework for characterizing alignment. Technical report, Knowledge Web project EU-IST-2004-507482, realizing the semantic web (2004)
3. Plato: The Republic. (360BC)
4. Gambetta, D.: Can we trust trust. In Gambetta, D., ed.: Trust: Making and Breaking Cooperative Relations, Basil Blackwell (1988) 213–237

5. Dasgupta, P.: Trust as a commodity. In Gambetta, D., ed.: Trust: Making and Breaking Cooperative Relations, Blackwell (1988) 49–72
6. Bromley, D.B.: Reputation, Image and Impression Management. John Wiley & Sons (1993)
7. Celentani, M., Fudenberg, D., Levine, D.K., Psendorfer, W.: Maintaining a reputation against a long-lived opponent. Econometrica **64**(3) (1966) 691–704
8. Sabater, J., Sierra, C.: Review on computational trust and reputation models. Artificial Intelligence Review **24**(1) (2005) 33–60
9. Ramchurn, S.D., Huynh, D., Jennings, N.R.: Trust in multi-agent systems. The Knowledge Engineering Review **19**(1) (2004) 1–25
10. Conte, R., Paolucci, M.: Reputation in Artificial Societies: Social beliefs for social order. Kluwer Academic Publishers (2002)
11. Omidyar, P.: Ebay. http://www.ebay.com, retrieved September 26, 2008 (1995)
12. Reiley, D., Bryan, D., Prasad, N., Reeves, D.: Pennies from ebay: The determinants of price in online auctions. Journal of Industrial Economics **55**(2) (2007) 223–233
13. Grice, P.: Logic and conversation. In Cole, P., Morgan, J.L., eds.: Syntax and Semantics, Vol. 3: Speech Acts. Academic Press, New York (1975) 41–58
14. Orsolini, M.: "dwarfs do not shoot": An analysis of children's justifications. Cognition and Instruction **11**(3) (1993) 281–297
15. Veneziano, E., Sinclair, H.: Functional changes in early child language: the appearance of references to the past and of explanations. Journal of Child Language **22**(3) (1995) 557–581
16. Hexmoor, H., Castelfranchi, C., Falcone, R.: Agent Autonomy. Kluwer Academic Publishers (2003)
17. Chandler, D.: Semiotics: The Basics. Routledge (2002)
18. Steels, L.: Semiotic dynamics for embodied agents. IEEE Intelligent Systems **21**(3) (May 2006) 32–38
19. Abdul-Rahman, A., Hailes, S.: Supporting trust in virtual communities. Proceedings of the 33rd Hawaii International Conference on System Sciences **6** (2000) 4–7
20. Pinyol, I., Sabater-Mir, J., Cuni, G.: How to talk about reputation using a common ontology: From definition to implementation. In: Proc. of Tenth Workshop "Trust in Agent Societies" at AAMAS '07, Honolulu, Hawaii, USA (2007) 90–102
21. Casare, S., Sichman, J.: Towards a functional ontology of reputation. In: AAMAS '05: Proc. of the fourth international joint conference on Autonomous Agents and Multiagent Systems, Utrecht, The Netherlands, ACM (2005) 505–511
22. Nardin, L.G., Brandão, A.A.F., Muller, G., Sichman, J.S.: Effects of expressiveness and heterogeneity of reputation models in the art-testbed: Some preliminar experiments using the soari architecture. In: Proc. of the Twelfth Workshop "Trust in Agent Societies" at AAMAS '09, Budapest, Hungary (2009)
23. Sabater-Mir, J., Paolucci, M., Conte, R.: Repage: REPutation and imAGE among limited autonomous partners. JASSS - Journal of Artificial Societies and Social Simulation **9**(2) (2006)
24. Hübner, J.F., Lorini, E., Herzig, A., Vercouter, L.: From cognitive trust theories to computational trust. In: Proc. of the Twelfth Workshop "Trust in Agent Societies" at AAMAS '09, Budapest, Hungary (2009) 55–67
25. Şensoy, M., Yolum, P.: A context-aware approach for service selection using ontologies. In: Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'06), Hakkodate, Hokkaido, Japan, IFAAMAS (2006) 931–938

26. Pinyol, I., Sabater-Mir, J.: Arguing about reputation. the lrep language. In Artikis, A., O'Hare, G., Stathis, K., Vouros, G., eds.: Engineering Societies in the Agents World VIII: 8th International Workshop, ESAW 2007. Volume 4995 of LNAI., Springer Verlag (2007) 284–299
27. Stranders, R., de Weerdt, M., Witteveen, C.: Fuzzy argumentation for trust. In Sadri, F., Satoh, K., eds.: Computational Logic in Multi-Agent Systems: 8th International Workshop, CLIMA VIII. Volume 5056 of LNCS., Springer Verlag (2008) 214–230
28. Jøsang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. Decision Support Systems **43**(2) (2007) 618–644
29. Teacy, W.T.L., Patel, J., Jennings, N.R., Luck, M.: Travos: Trust and reputation in the context of inaccurate information sources. Autonomous Agents and Multi-Agent Systems **12**(2) (2006) 183–198
30. Jøsang, A., Ismail, R.: The beta reputation system. In: Proceedings of the Fifteenth Bled Electronic Commerce Conference e-Reality: Constructing the e-Economy, Bled, Slovenia (2002)
31. Şensoy, M., Zhang, J., Yolum, P., Cohen, R.: Context-aware service selection under deception. Computational Intelligence **25**(4) (2009) 335–366
32. Barwise, J., Seligman, J.: Information Flow: The Logic of Distributed Systems. Cambridge University Press (1997)
33. Schorlemmer, M., Kalfoglou, Y., Atencia, M.: A formal foundation for ontology-alignment interaction models. International Journal on Semantic Web and Information Systems **3**(2) (2007) 50–68
34. Atencia, M., Schorlemmer, M.: A formal model for situated semantic alignment. In: Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2007), Honolulu, Hawaii, USA, IFAAMAS (2007) 1270–1277
35. Koster, A., Sabater-Mir, J., Schorlemmer, M.: A formalization of trust alignment. In Sandri, S., Sànchez-Marré, M., Cortes, U., eds.: AI Research and Development. Proc. of the Twelfth International Congress of the Catalan Association of Artificial Intelligence (CCIA 2009). Volume 202 of Frontiers in Aritficial Intelligence and Applications., Cardona, Spain, IOS Press (2009) 169–178
36. De Raedt, L.: Logical and Relational Learning. Springer Verlag (2008)
37. Koster, A., Sabater-Mir, J., Schorlemmer, M.: Inductively generated trust alignments based on shared interactions (extended abstract). In: Ninth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2010), Toronto, Canada, IFAAMAS (In Press)
38. Koster, A., Sabater-Mir, J., Schorlemmer, M.: An interaction-oriented model of trust alignment. In: Proc. of the 13th Conference of the Spanish Association for Artificial Intelligence, CAEPIA 2009, Sevilla, Spain (2009) 655–664
39. Urbano, J., Rocha, A.P., Oliveira, E.: A trust aggregation engine that uses contextual information. In: Proc. of EUMAS 2009. (2009)
40. Hermoso, R., Billhardt, H., Ossowski, S.: Dynamic evolution of role taxonomies through multidimensional clustering in multiagent organizations. In: Proc. of EUMAS 2009. (2009)
41. Ramon, J., Bruynooghe, M.: A polynomial time computable metric between point sets. Acta Informatica **37** (2001) 765–780
42. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Communications **7**(1) (1994) 39–59

# False Name Manipulations in Weighted Voting Games: Susceptibility of Power Indices

Ramoni O. Lasisi and Vicki H. Allan

Department of Computer Science, Utah State University,
Logan, UT 84322-4205, USA
ramoni.lasisi@aggiemail.usu.edu,vicki.allan@usu.edu

**Abstract.** The splitting of weights into smaller sizes by agents in a weighted voting game and the distribution of the new weights among several false identities with the intent of payoff or power increase in a new game consisting of the original agents as well as the false identities is called *false name manipulation.* In this paper, we study false name manipulations in weighted voting games focusing on the power indices used in evaluating agents' payoff in such games. We evaluate the susceptibility to false name manipulations in weighted voting games of the following power indices, namely, Shapley-Shubik, Banzhaf and Deegan-Packel indices when an agent splits into several false identities. Our experimental results suggest that the three power indices are susceptible to false name manipulations when an agent splits into several false identities. However, the Deegan-Packel power index is more susceptible than Shapley-Shubik and Banzhaf indices.
**General Terms:** Algorithms, Economics, Theory

**Key words:** Agents, Manipulations, Power indicies, Trust

## 1 Introduction

Cooperation among self-interested autonomous agents in multiagent environments is fundamental for agents to successfully achieve goals for which they lack enough resources and skills. The level of skills and amount of resources of agents varies, hence the need for agents' cooperation to complete tasks that are otherwise difficult for individual agents to achieve or for which better results (than working independently) can be attained. One way of modelling such cooperation is via *weighted voting games.*

Weighted voting games (WVGs) are mathematical abstractions of voting systems. In a voting system, voters express their opinions through their votes by electing candidates to represent them or influence the passage of bills. Each member voters, $V$, has an associated weight $w : V \to Q^+$. A voter's weight is the number of votes controlled by the voter, and this is the maximum number of votes she is permitted to cast. The *homogeneous voting system* is a special case in which all voters have unit weight [7]. In our context, a subset of agents, called the *coalition*, wins in a WVG, if the sum of the weights of the individual agents in the coalition meets or exceeds a certain threshold called the *quota*. In the case of the more traditional homogeneous voting system, the winning coalition (WC) is determined by the majority of the agents. On the other hand, in the usual WVGs with all agents having different weights, a coalition with sum of the individual agents' weights meeting or exceeding the quota determines the WC.

It is natural to naively think that the numerical weight of an agent directly determines the corresponding strength of the agent in a WVG. The measure of the strength of an agent is its *power*. This is the ability of an agent to influence the decision-making process. Consider, for example, a WVG of three voters, $a_1, a_2$, and $a_3$ with respective weights $6, 3$, and 1. Suppose the quota for the game is 10, then it is clear that a coalition consisting of all the three voters is needed to win the game. Thus, each of the voters $a_1, a_2$, and $a_3$ are of equal importance in achieving the WC. Hence, they each have equal power irrespective of their weight distribution in that every voter is necessary for a win.

A strategic agent may alter a game in anticipation of power increase by splitting its weight among several false agents that are not in the original game. Bachrach and Elkind [3] refer to this action as *false-name manipulation*. The new game consists of all the previous agents and the several false identities into which the manipulating agent splits. The power of the agent is thus the sum of the powers of all its false identities. This agent anticipates that the value of its accumulative power to be at least the value in the original game. Bachrach and Elkind [3] and Aziz and Paterson [1] show that this anticipation of power increase due to splitting into exactly two false identities is not achieved at all times. There are cases when the cumulative power of the false identities remains the same or even decreases compared with the original power of the agent.

Common measures of agents' power are the Shapley-Shubik, Banzhaf, and Deegan-Packel power indices [8]. Bachrach and Elkind [3] and Aziz and Paterson [1] evaluate the effects of false name manipulation when an agent splits into exactly two false identities using Shapley-Shubik and Banzhaf indices respectively.

To date, there has been practically no work on the effect of false name manipulations when an agent splits into more than two identities and thus, remains unexplored [1]. In this paper, we evaluate the susceptibility to false name manipulations in WVGs of the following power indices, namely, Shapley-Shubik, Banzhaf, and Deegan-Packel indices for the case when an agent splits into several false identities. The more resistant to manipulation a power index is the better. Hence, agents' motivation towards manipulation is thus reduced. This provides some assurance of identity, which is crucial for establishing and maintaining trustworthy interactions. The remainder of the paper is organized as follows. Section 2 discusses related work. Section 3 provides the definitions and notations used in the paper. In Sect. 4, we provide examples to illustrate false name manipulation when an agent splits into exactly two false identities using the Deegan-Packel index. In Sect. 5, we provide experimental evaluation of susceptibility of the three power indices to false name manipulations. We conclude in Sect. 6 and provide directions for future work.

## 2 Related Work

WVGs and power indices are widely studied [1],[3],[4],[8]. WVGs have many applications, including economics, political science, neuroscience, threshold logic, reliability theory, distributed systems [2], and multiagent systems [3]. Aziz et al. provide a brief discussion of some applications of WVGs. Prominent instances of weighted voting problems are in the United Nations Security Council, the Electoral College of the United States and the International Monetary Fund [2].

The study of WVGs has also necessitated the need to fairly determine the strength or power of players in a game. This is because the power of a player in a game provides information about the relative importance or criticality of that player in the game compared to other players. To evaluate the power of the players, power indices such as Shapley-Shubik, Banzhaf, and Deegan-Packel indices are commonly employed [8]. These power indices satisfy the axioms that characterize a power index [3], have gained wide usage in political arena, and are the main power indices found in the literature [6]. Computing the power indices of players using any of Shapley-Shubik, Banzhaf index, or Deegan-Packel index is NP-hard [8], and the problem is also #P-complete for Shapley-Shubik and Banzhaf. However, these values can be computed in pseudo-polynomial time by dynamic programming [5], [8]. There are also approximation algorithms for computing the power indices using Shapley-Shubik and Banzhaf indices [4].

These power indices have been defined on the framework of subsets of WCs in the game they seek to evaluate. A wide variation in the results they provide can be observed. This is due partly to the different definitions and methods of computation of the associated subsets of the WCs. Then, comes the question of which of the power indices is the most resistant to manipulation in a WVG. The choice of a power index depends on a number of factors, namely, the a priori properties of the index, the axioms characterizing the power indices, and the context of decision making process under consideration [6].

False name manipulation has been studied in the context of non-cooperative games [3] and in open anonymous environments, such as the internet [9]. False name manipulation is hard to discover and can be effective in such environments. The menace can take different forms, such as agents providing multiple identities, two or more agents merging identities to form a single agent, non disclosure of full status (in the form of hiding skills) by agents or even a combination of these forms [9]. The maiden study of this behavior in the context of WVG is the work of Bachrach and Elkind [3]. This action involves an agent splitting into a number of false agents with the intent that the cumulative power index value of the false agents exceeds the original value. They use the Shapley-Shubik index to evaluate agent power and consider the case when agents splits into exactly two false agents. The extent to which agents increase or decrease their Shapley power are also bounded. Similar results using the Banzhaf power index were obtained by Aziz and Paterson [1].

## 3 Definitions and Notations

We give the following definitions and notations used throughout the paper.

**Weighted Voting Game**: Let I $= \{1, \cdots, n\}$ be a set of agents. Let $\mathbf{w} = \{w_1, \cdots, w_n\}$ be the corresponding positive integer weights of the agents in order. Let $S$ be a non empty set of agents. $S \subseteq I$ is a coalition. A WVG $G$ with quota $q$ involving agents $I$ is defined as $G = [w_1, \cdots, w_n; q]$. Denote by $w(S)$, the weight of a coalition $S$ derived from the summation of the individual weights of agents in $S$ i.e. $w(S) = \sum_{i \in S} w_i$. A coalition, $S$, wins in the game $G$ if $w(S) \geq q$ otherwise it loses. So that simultaneously there can be a single WC, $q$ is constrained as follows $\frac{1}{2}w(I) < q \leq w(I)$.

**Simple Voting Game**: Each of the $2^{|I|}$ coalitions $S \subseteq I$ has an associated function $v : S \rightarrow \{0, 1\}$. The value 1 implies a win for the coalition and 0 a loss. In the game $G, v(S) = 1$ if $w(S) \geq q$ and 0 otherwise.

**Dummy and Critical Agents**: An agent $i \in S$ is *dummy* if its weight in $S$ is not needed for $S$ to be a WC, i.e. $w(S - \{i\}) \geq q$. Otherwise it is *critical* to coalition $S$, i.e. $w(S) \geq q$ and $w(S - \{i\}) < q$.

**Unanimity Weighted Voting Game**: A WVG in which there is a single WC and every agent is critical to the coalition is a *unanimity weighted voting game.*

**Shapley-Shubik Power Index**: The Shapley-Shubik power index is one of the oldest power indices and has been used widely to analyze political power. The index quantifies the marginal contribution of an agent to the grand coalition. Each agent in a permutation is given credit for the win if the agents preceding it do not form a WC but by adding the agent in question, a WC is formed. The power index is dependent on the number of permutations for which an agent is critical. For the $n!$ permutations of agents used in determining the Shapley-Shubik index, there exists exactly one critical agent in each of the permutations. Adopting Bachrach and Elkind's notation [3], we denote by $\Pi$ the set of all permutations of $n$ agents in a WVG G. Let $\pi \in \Pi$ define a one-to-one mapping where $\pi(i)$ is the position of the $i$th agent in the permutation order. Denote by $S_\pi(i)$, the predecessors of agent $i$ in $\pi$, i.e., $S_\pi(i) = \{j : \pi(j) < \pi(i)\}$. The Shapley-Shubik value of the $i$th agent in $G$ is given by

$$\varphi_i(G) = \frac{1}{n!} \sum_{\pi \in \Pi} [v(S_\pi(i) \cup \{i\}) - v(S_\pi(i))] \qquad (1)$$

**Banzhaf Power Index**: Another index that has also gained wide usage in the political arena is the Banzhaf power index. Unlike the Shapley-Shubik index, its computation depends on the number of WCs in which an agent is critical. There can be more than one critical agent in a particular WC. The Banzhaf index, $\beta_i(G)$, of agent $i$ in the same game, $G$, as above is given by

$$\beta_i(G) = \frac{\eta_i(G)}{\sum_{i \in I} \eta_i(G)} \qquad (2)$$

where $\eta_i(G)$ is the number of coalitions in which $i$ is critical in $G$.

**Deegan-Packel Power Index**: The Deegan-Packel power index is also found in the literature for computing power indices. The computation of this power index for an agent $i$ takes into account both the number of all the minimal winning coalitions (MWCs) in the game as well as the sizes of the MWCs having $i$ as a member [8]. Thus, it is more impressive to be one in three (who elicited the win) rather than one in ten. A WC $C \subseteq I$ is a MWC if every proper subset of $C$ is a losing coalition, i.e. $w(C) \geq q$ and $\forall T \subset C, w(T) < q$. The Deegan-Packel power index, $\gamma_i(G)$, of an agent $i$ in $G$ is given by

$$\gamma_i(G) = \frac{1}{|MWC|} \sum_{S \in MWC_i} \frac{1}{|S|} \tag{3}$$

where MWC$_i$ are the sets of all MWCs in $G$ that include $i$.

**Susceptibility of Power Index to Manipulation**: Let $\Phi$ be a power index. Denote by $\Phi_i(G)$, the power of an agent $i$ in a WVG $G$. Suppose $i$ alters $G$ by splitting into $k$ false identities having weights $w_{i1}, \cdots, w_{ik}$. Let $G'$ be the resulting game such that $\sum_{j=1}^{k} w_{ij} = w_i$ and the weights of all other agents in $G'$ remain constant. We say that $\Phi$ is susceptible to manipulation if there exists $\sum_{j=1}^{k} \Phi_{ij}(G') > \Phi_i(G)$, and the split is *advantageous.* If $\sum_{j=1}^{k} \Phi_{ij}(G') < \Phi_i(G)$, then the split is *disadvantageous* while it is *neutral* when $\sum_{j=1}^{k} \Phi_{ij}(G') = \Phi_i(G)$.

## 4 False Name Manipulations with Deegan-Packel Index

In this section, we provide examples to illustrate false name manipulation in WVGs using the Deegan-Packel power index to compute power. We consider the case where an agent splits into exactly two false identities in the new game. Splitting into more than two false identities is considered in the next section.

*Example 1.* Splitting Advantageous

Let $G = [5, 4, 3; 7]$ be a WVG of three agents $1, 2,$ and $3$ having respective weights $5, 4,$ and $3$ with quota $q = 7$. The game has three MWCs $\{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$. Consider agent 1. The Deegan-Packel index of this agent computed using (3) above is $\gamma_1(G) = \frac{1}{3}$. Suppose the agent splits into two new false agents $1_a$ and $1_b$ with respective weights 3 and 2. We have a new game $G' = [3, 2, 4, 3; 7]$. The MWCs for this game consist of $\{\{1_a, 2\}, \{1_a, 1_b, 3\}, \{2, 3\}\}$. The respective Deegan-Packel indices for agents $1_a$ and $1_b$ are $\gamma_{1_a}(G') = \frac{5}{18}$ and $\gamma_{1_b}(G') = \frac{1}{9}$. Clearly, the sum of the values of the two indices, namely $\gamma_{1_a}(G') + \gamma_{1_b}(G') = \frac{7}{18} > \gamma_1(G)$. Thus, the agent benefits from the split action by an increase in payoff.

*Example 2.* Splitting Disadvantageous

Let $G = [20, 10, 8, 8, 3, 2, 1, 1; 28]$ be a WVG. The Deegan-Packel index of agent 1 is $\gamma_1(G) = 0.2500$. Suppose this agent splits into $1_a$ and $1_b$ with weights 14 and 6 respectively in a new game $G' = [14, 6, 10, 8, 8, 3, 2, 1, 1; 28]$. The Deegan-Packel indices of agents $1_a$ and $1_b$ are 0.1339 and 0.1113 respectively. Hence, $\gamma_{1_a}(G') + \gamma_{1_b}(G') = 0.2452 < \gamma_1(G)$.

*Example 3.* Splitting Neutral

Let $G = [5, 4, 3; 7]$ be the same WVG as example 1. Suppose the agent splits into $1_a$ and $1_b$ with respective weights 4 and 1 in a new game $G' = [4, 1, 4, 3; 7]$. The Deegan-Packel indices of agents $1_a$ and $1_b$ are $\frac{1}{3}$ and 0 respectively. Hence, $\gamma_{1_a}(G') + \gamma_{1_b}(G') = \gamma_1(G)$, and the agent neither benefited nor incured a decrease in payoff.

## 5 Susceptibility of Power Indices to Manipulations

In this section, we demonstrate the susceptibility to false name manipulations in WVGs of the following power indices, namely, Shapley-Shubik, Banzhaf and Deegan-Packel power indices. We consider the more general case of agents splitting into more than two identities. For the sake of simplicity in our discussion, we provide the following assumptions that by no means invalidate the basic requirements of WVGs:

1. Only one of the agents is engaging in the manipulation at a time. We note that other agents also have similar motivation to split their weights in anticipation of power increase. In our future work, we intend to consider scenarios where multiple agents in WVGs simultaneously engage in false name manipulation.
2. All splits of agents' weights are into integer values. Otherwise we will have an exponential number of possible new weights to evaluate.
3. The weights of the false identities that an agent splits into are strictly greater than zero. By the null player axiom [6], assigning a weight of zero to an agent does not make the agent critical in all WCs.

### 5.1 Unanimity Weighted Voting Games

We recall that a WVG in which there is a single WC and such that every agent is critical to the coalition is a unanimity WVG game. Since all the agents in the WC of unanimity WVGs are critical, the total weights of all agents, $w(I)$ and the quota, $q$ in such games satisfy the inequality, $w(I) \geq q$.

**Proposition 1.** *In a unanimity WVG with $q = w(I)$, if Banzhaf indices are used as payoffs of agents in a WVG, then it is beneficial for an agent to split up into several agents. The same holds for Shapley-Shubik power index [1].*

**Proposition 2.** *In a unanimity WVG with $q = w(I)$, if the Deegan-Packel index is used to compute power of agents, then it is advantageous for an agent to split up into several false agents.*

*Proof.* Let $G$ be a unanimity WVG of $n$ agents with quota $q = w(I)$. It is easy to see that the Deegan-Packel power index of every agent $i$ in $G$, $\gamma_i(G) = \frac{1}{n}$. Suppose agent 1 splits into $m + 1$ false agents, then we have a new unanimity game, $G'$ of $n + m$ agents. The Deegan-Packel power index of every agent $i$ in $G'$,$\gamma_i(G') = \frac{1}{n+m}$. Hence, the new Deegan-Packel power index of agent 1 is $\gamma_1(G') = \frac{m+1}{n+m} > \frac{1}{n}$ for $n > 1$. □

The following theorem is immediate from propositions 1 and 2.

**Theorem 1.** *Let $G$ be a unanimity WVG of $n$ players with quota $q = w(I)$. Suppose an agent $i$ splits into $k \geq 3$ false agents. The Shapley-Shubik, Banzhaf, and Deegan-Packel power index of agent $i$ increases as its split size, $k$, increases.*

**Corollary 1.** *Let $G$ be a unanimity WVG with $w(I) > q$. Let an agent $i$ split into several false agents in a new game $G'$. Suppose the new game $G'$ is also a unanimity WVG, then the splitting is advantageous for $i$ if any of Shapley-Shubik, Banzhaf, and Deegan-Packel power index is use to compute the agent's power.*

### 5.2 General Weighted Voting Games

False name manipulation in the general case of the WVGs is more interesting as it provides more complex and realistic scenarios that are not well-understood. Of importance is that the number of WCs and MWCs changes in contrast to being static as observed with unanimity WVGs with $q = w(I)$. Thus, as the structure of the WVGs changes, so does the composition of the WCs and MWCs.

As mentioned in the introduction, the more detailed analysis on the effect of false name manipulations when an agent splits into more than two false agents remain unexplored [1]. The only closely related research are the NP-hardness results of finding a beneficial split in WVGs when a manipulating agent splits into at least two false agents. The hardness results are for the Shapley-Shubik [3] and Banzhaf [1] power indices. To the best of our knowledge, ours is the first paper to confirm the existence of beneficial splits when agents split into more than two false identities for the three power indices we consider.

We perform experiments to simulate the effect of manipulations by agents using the three power indices. The weights of our agents are chosen so that no weight is larger than ten. These weights are reflective of realistic voting procedures as the weights of agents in real voting are not too large [3] and as such are representative of WVGs. In the experiments, we randomly generate WVGs and assume only the first agent in the game is engaging in the manipulation, then determine the three power index values of this agent in the game. After this, we consider splits into at least two false identities by this agent while the weights of all other agents remain the same in the new games. Suppose the initial weight of the manipulating agent in the original game is $n$, we allow the agent, to split its weight among the false agents in the new games as follows, $\{\{n-1,1\},\{n-2,1,1\},\cdots,\{1,1,\cdots,1\}\}$. The values of the power indices of the several false agents into which the manipulating agent splits are then added and compared with the power index of the agent in the original game. We generate 20,000 original WVGs for the experiments and allowed the manipulating agent to split its weight in each of the games. The numbers of the new games generated by the action of the manipulating agent depends on the initial weight of the agent in the original games.

Our experiments suggest the existence of beneficial splits when agents engage in such manipulations for the three power indices. However, the extent to which agents gain varies with the indices. The effect of this action is well noticed with the Deegan-Packel index as we found cases where agents improve their power index by more than four times the original value. On the other hand, the maximum gain attained while using any of Shapley-Shubik and Banzhaf index is less than a factor of two. The result suggests that Deegan-Packel power index is more susceptible to false name manipulations than the Shapley-Shubik and the Banzhaf indices. Hence, this may provides some motivation for agents to engage in manipulation in WVGs when the game is being evaluated with the Deegan-Packel index. We illustrate three games from the experiments in which an agent attains high factor splitting into more than two false agents for the three power indices.

*Example 4.* Consider the WVG $G = [6, 2, 2, 3, 10; 12]$. The gains of the manipulating agent (with weight 6) is depicted in Table 1 for the Shapley-Shubik power index.

**Table 1.** The splitting agent weight, Shapley-Shubik indices, and the factor of increment in the game $G = [6, 2, 2, 3, 10; 12]$.

| Splitting Agent Weights | Shapley-Shubik Index | Factor Increment |
|---|---|---|
| $\{6\}$ | 0.1000 | - |
| $\{5, 1\}$ | 0.1000 | - |
| $\{4, 1, 1\}$ | 0.1238 | 1.2 |
| $\{3, 1, 1, 1\}$ | 0.1429 | 1.4 |
| $\{2, 1, 1, 1, 1\}$ | 0.1548 | 1.5 |
| $\{1, 1, 1, 1, 1, 1\}$ | 0.1667 | 1.7 |

*Example 5.* Consider the WVG $G = [7, 8, 4, 8, 4; 16]$. The gains of the manipulating agent (with weight 7) is depicted in Table 2 for the Banzhaf power index.

**Table 2.** The splitting agent weight, Banzhaf indices, and the factor of increment in the game $G = [7, 8, 4, 8, 4; 16]$.

| Splitting Agent Weights | Banzhaf Index | Factor Increment |
|---|---|---|
| $\{7\}$ | 0.1429 | - |
| $\{6, 1\}$ | 0.1429 | - |
| $\{5, 1, 1\}$ | 0.1429 | - |
| $\{4, 1, 1, 1\}$ | 0.1429 | - |
| $\{3, 1, 1, 1, 1\}$ | 0.1864 | 1.3 |
| $\{2, 1, 1, 1, 1, 1\}$ | 0.2381 | 1.7 |
| $\{1, 1, 1, 1, 1, 1, 1\}$ | 0.2672 | 1.9 |

*Example 6.* Consider the WVG $G = [8, 4, 9, 1, 4; 14]$. The gains of the manipulating agent (with weight 8) is depicted in Table 3 for the Deegan-Packel power index.

**Table 3.** The splitting agent weight, Deegan-Packel indices, the factor of increment, and the number of MWCs in the game $G = [8, 4, 9, 1, 4; 14]$.

| Splitting Agent Weights | Deegan-Packel Index | Factor Increment | # of MWC |
|---|---|---|---|
| $\{8\}$ | 0.1667 | - | 5 |
| $\{7, 1\}$ | 0.2143 | 1.3 | 7 |
| $\{6, 1, 1\}$ | 0.2407 | 1.4 | 9 |
| $\{5, 1, 1, 1\}$ | 0.3036 | 1.8 | 14 |
| $\{4, 1, 1, 1, 1\}$ | 0.4207 | 2.5 | 29 |
| $\{3, 1, 1, 1, 1, 1\}$ | 0.5344 | 3.2 | 57 |
| $\{2, 1, 1, 1, 1, 1, 1\}$ | 0.6201 | 3.7 | 115 |
| $\{1, 1, 1, 1, 1, 1, 1, 1\}$ | 0.6754 | 4.1 | 229 |

For this example, the number of MWCs in the original game is 5. They are $\{3, 4, 5\}, \{2, 3, 5\}, \{2, 3, 4\}, \{1, 3\}$, and $\{1, 2, 5\}$. Although the weight of agent 1, 8 is relatively high compared to some other agents in the game, it belongs to only two of the MWCs having two and three members. Of particular interest are agents 2 and 5 with weight 4 each, these agents belong to three MWCs with each of the coalitions having three members, hence, by (3), their power from the original game is greater than that of agent 1 with higher weight. Agent 1 thus has motivation to split its weight. So, as the agent splits its weight, it becomes active in more MWCs which improves its power index.

### 5.3 Simulation Results

We present the results of our extensive set of simulations. For our study, we generate 20,000 original WVGs and allow manipulation of all the games by the manipulating agent. When starting a new game, all agents are randomly assigned weights in the current game and the quota of the game is also generated based on the weights assumed by the agents. We designate the first agent to be the manipulating agent. We have five agents in each of the original games. The maximum weight that can be asssumed by the manipulating agent is eight while all other agents can be up to ten. The least possible weights for any agent is one. Obviously, when the manipulating agent assumes a weight of one in a game, then it is not possible for it to split in such game. We keep the weight of the manipulating agent to be lower than other agents to limit the number of cases. Since all weigths are randomly generated, we have a handfull of different types of games that are representative of the WVGs. For example, we have many cases where the original weight of the manipulating agent is lower, higher or even the same as the weights of many or all agents in the original games.

We first consider how susceptibility to manipulation among the power indices compares when an agent is allowed to manipulate a game. This is achieved by comparing the population of the factor of increment attained by agents in different games for each indices with the split sizes (the number of false agents the original agent splits). We show a summary of the ease of manipulation by agents for the three indices in $20,000$ WVGs in Fig. 1. The x-axis indicates the split size while the y-axis is the average factor of increment achieved by agents in the $20,000$ WVGs for different split sizes. The high susceptibility of the Deegan-Packel index to manipulation can be observed from the figure. While the average factor of increment for manipulation rapidly grows with split size for this index, the growth for the Shapley-Shubik and Banzhaf does not appear to correlate with split sizes, and on average does not improve utility. From our experiments, many of the games are advantageous with the Deegan-Packel index while many are disadvantageous for both Shapley-Shubik and Banzhaf indices. This result is indicative of the ease by which each of the power indices is manipulable.

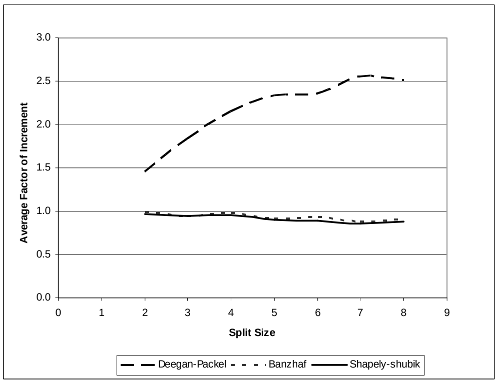


**Fig. 1.** Ease of Manipulation among the three Power Indices

The number of games that are advantageous, neutral, and disadvantageous for the three indices are also analyzed. Figure 2 shows that a larger number of the games are advantageous for Deegan-Packel index than for Shapley-Shubik and Banzhaf indices. Consider when $1,000$ games are generated, in more than 800 of the games are splitting advantageous for Deegan-Packel while we have less than 300 advantageous games for Shapley-Shubik and Banzhaf indices. Similarly, Fig. 3 shows the number of neutral games. We have more neutral games for Shapley-Shubik and Banzhaf than Deegan-Packel. Virtually none of the games are neutral for Deegan-Packel while about 90 of the games are neutral for Shapley-Shubik and Banzhaf indices out of a collection of $1,000$ games. Finally, Fig. 4 shows that there are fewer disadvantageous games for Deegan-Packel compare with Shapley-Shubik and Banzhaf indices. Clearly, the Deegan-Packel index is more susceptible to false name manipulations than Shapley-Shubik and Banzhaf indices.

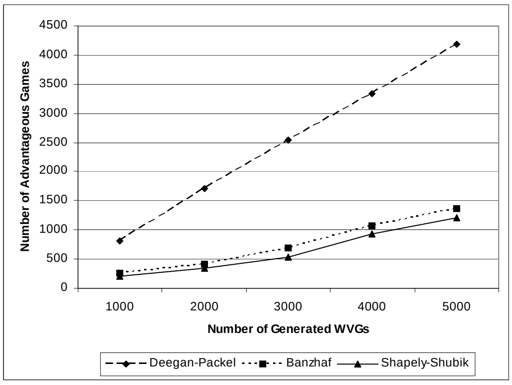


**Fig. 2.** Number of Advantageous Games among the Generated WVGs

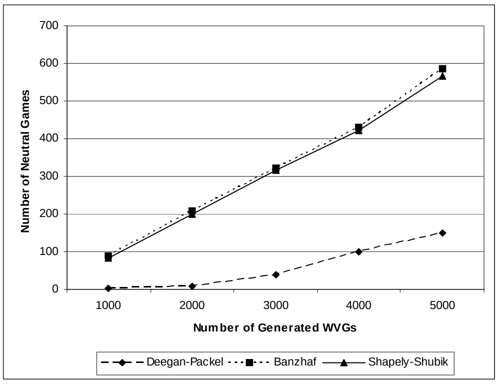


**Fig. 3.** Number of Neutral Games among the Generated WVGs

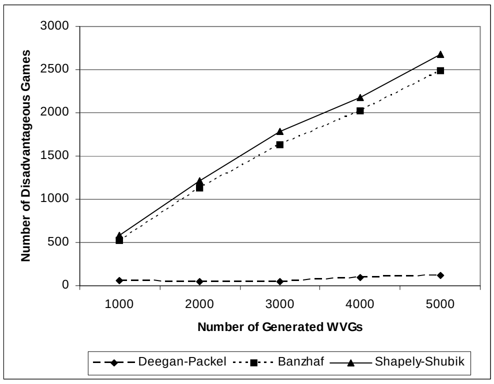


**Fig. 4.** Number of Disadvantageous Games among the Generated WVGs

## 6 Conclusions

In this paper, we evaluate the susceptibility to false name manipulations of the following power indices, namely, Shapley-Shubik, Banzhaf, and Deegan-Packel indices when an agent splits into several false identities in weighted voting games. We illustrate the susceptibility of the three indices through simulations of a large number of weighted voting games with a manipulating agent in each of the games. Our experimental results suggest that the three indices are susceptible to false name manipulations when an agent splits into several false identities. However, the Deegan-Packel index is more susceptible than Shapley-Shubik and Banzhaf indices, with Shapley-Shubik being the least susceptible. Hence, using Shapley-Shubik index to evaluate weighted voting games reduces agents' motivation towards false name manipulations. This provides some assurance of identity, which is crucial for establishing and maintaining trustworthy interactions.

Since our experimental results have suggested ideas on the extent to which each of the three indices are susceptible to false name manipulations, an obvious direction for future work is to provide theoretical bounds on the extent to which each of the indices are susceptible to false name manipulations when an agent splits into several false identities. It will also be interesting to come up with desirable properties that power indices should satisfy in order to prevent false name manipulations or prove that such properties are not achievable.

## References

1. Aziz, H., Paterson, M.: False-name Manipulations in Weighted Voting Games: splitting, merging and annexation. In: 8th International Conference on Autonomous Agents and Multiagent Systems, pp. 409–416. Budapest, Hungary (2009)
2. Aziz, H., Paterson, M., Leech, D.: Combinatorial and Computational Aspects of Multiple Weighted Voting Games. The Warwick Economics Research Paper Series (TWERPS) 823, University of Warwick, Department of Economics (2007)
3. Bachrach, Y., Elkind, E.: Divide and Conquer: False-name Manipulations in Weighted Voting Games. In: 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008), pp. 975–982. Estoril, Portugal (2008)
4. Bachrach, Y., Markakis, E., Procaccia, A.D., Rosenschein, J.S., Saberi, A.: Approximating Power Indices. In: 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008), pp. 943–950. Estoril, Portugal (2008)
5. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman, San Fransisco (1979)
6. Laruelle, A.: On the Choice of a Power Index. Instituto Valenciano de Investigaciones Economicas. 2103, 99–10 (1999)
7. Levchenkova, L.G., Levchenkov, V.S.: A Power Index in Weighted Voting Systems. Journal of Computational Mathematics and Modeling. 13(4), 375–392 (2002)
8. Matsui, T., Matsui, Y.: A Survey of Algorithms for Calculating Power Indices of Weighted Majority Games. Journal of the Operations Research Society of Japan, 43(1), 2000
9. Yokoo, M., Conitzer, V., Sandholm, T., Ohta, N., Iwasaki, A.: Coalitional Games in Open Anonymous Environments. In: American Association for Intelligence Conference. pp. 509–515 (2005)

# A Propagation and Aggregation Algorithm for Inferring Opinions in Structural Graphs

Nardine Osman, Carles Sierra, and Angela Fabregues

Artificial Intelligence Research Institute (IIIA-CSIC), Barcelona, Spain

**Abstract.** This paper is concerned with the issue of reputation and the computation of group opinion. We argue that entities may receive both objective and subjective opinions, and distinguishing between the two is crucial for achieving more precise measures. Additionally, we argue that the group opinion about an entity $\alpha$ is not only influenced by the opinions that $\alpha$ receives (whether objective or subjective), but by the reputation of other entities that $\alpha$ is related to. As such, we propose a method that permits the propagation and aggregation of opinions in structural graphs, allowing the inference of more precise reputation measures through the description of both objective and subjective group opinion.

**Keywords:** Trust, reliability, reputation, opinion propagation & aggregation

## 1 Introduction

There is a common understanding that reputation represents group opinion. Existing work has mainly focused on the direct opinions an entity (whether a person, a peer, an agent, or even an item) receives. This paper, however, introduces the concept of inferred opinions, based on relations between entities. Consider the example of having a poorly reputable football team that just started hiring well known players. Naturally, one would say that such a move is reasonable since it increases the team's chances, or *expectations*, of winning games, and hence, increase the team's reputation. These indirect opinions are highlighted when the entities are related. In other words, the reputation of different entities may influence each other when these entities are related. For example, having one entity being a part of another implies a *propagation* of opinions between these two entities. Hence, to consider indirect influential opinions, one should have a clear definition of the relations that link entities together, since opinions may only propagate along such relations. For the time being, we focus on the simple 'part of' relation. This results in the construction of a structural graph, which we define in the following section.

In addition to the introduction of the notion of opinion propagation in structural graphs, we also introduce the distinction between objective and subjective opinions. For example, the team being highly rated by some magazine should not be as influential in comparison with the team's actual performance, such as losing major games. We categorise opinions accordingly:

- **Objective opinions.** These opinions may not be falsified.[1] For example, if one ping pong player wins against another, then this may be interpreted as the former player stating that the latter is weaker, and vice versa.
- **Subjective opinions.** These are divided into two further subclasses.
  - *Direct subjective opinions.* For example, a scientific paper may win an award, or a scientific paper may receive good reviews from experts in the field. These may be viewed as direct subjective opinions.
  - *Influential opinions of related entities.* We say opinions may propagate from a parent node in the structural graph to its children nodes, and vice versa. For example, the final opinion about a scientific paper may be influenced by the final opinion about the conference where it has been accepted. Similarly, the opinion about a conference may also be influenced by the opinions about the papers it accepted.

The rest of this paper is divided as follows. Section 2 provides the basic definitions that the proposed algorithm is based on. The main model is then introduced by Section 3, which illustrates how group opinion may be calculated and how reputation is defined accordingly. The entire algorithm is summarised by Section 4. Some preliminary results are presented by Section 5, before Section 6 closes with a brief conclusion.

## 2 Basic Definitions

In what follows, we provide a clear definition of the structural graph that is needed for the propagation of opinions (Section 2.1), what direct individual opinions are (Section 2.2), and what group opinion is (Section 2.3).

### 2.1 The Structural Graph

We define a graph whose nodes represent entities that may form or receive opinions as follows:

**Definition 1.**

$$SG = \langle N, G, O, E, A, T, \mathcal{P}, \mathcal{F} \rangle$$

*where*

- *$N$ is the set of nodes, or entities,*
- *$G$ is the set of agents, peers, people, or even entities that may form opinions about $\alpha \in N$ (we note that $G$ and $N$ may or may not intersect, depending on the different fields of application),*
- *$O$ is the set of direct opinions, whose elements are defined shortly,*
- *$E = \{e_1, ..., e_n\}$ is the evaluation space for $O$, where terms $e_i$ account for terms like 'bad', 'good', 'very.good', etc.,*

[1] Objective information is not usually referred to as 'opinions'. However, to compute reputation, which is generally defined as group opinion, we propose to interpret all sources of information that could influence reputation as opinions.

- *$A$ is the set of attributes (e.g. $\{strength, quality, \dots\}$) that opinions address,*
- *$T$ represents calendar time,*
- *$\mathcal{P} \subseteq N \times N$ specifies which nodes are part of the structure of which others,*
- *$\mathcal{F} : R \times N \times A \times T \rightarrow O$ is a relation that links a given agent, node, attribute, and time to an opinion.*

### 2.2 Individual Opinions

We define an opinion $o^t_\alpha(\beta) \in O$ as follows:

**Definition 2.**

$$o^t_\alpha(\beta) = \{e_1 \mapsto v_1, ..., e_n \mapsto v_n\}$$

*where,*

- *$t \in T$, $\alpha \in G$, and $\beta \in N$,*
- *$\{e_1, ..., e_n\} = E$,*
- *$v_i \in [0, 1]$ represents the value assigned to each element $e_i \in E$, with the condition that $\sum_{v_i} v_i = 1$.*

In other words, $o^t_\alpha(\beta)$ represents the opinion that an entity $\alpha$ may hold about entity $\beta$ at time $t$; and the opinion is specified as a discrete probability distribution over the evaluation space $E$.[2] We note that the opinion one holds with respect to another may change with time, hence various instances of $o^t_\alpha(\beta)$ may exist for the same $\alpha$ and $\beta$ but with distinct $t$s.

### 2.3 Group Opinion

This paper distinguishes between two different types of group opinion, based on the categorisation of opinions presented by Section 1:

- Group opinion based on objective opinions: $\mathbb{D}^t_\alpha(e_i)$
- Group opinion based on subjective opinions: $\mathbb{P}^t_\alpha(e_i)$

[2] We note that this paper only considers a discrete evaluation space $E$. In our proposed algorithm, as the equations of this paper illustrate, the main operations that are carried out over the probability distributions are the application of the addition $+$, subtraction $-$, multiplication $\times$, and division $/$ operators. These operators could easily be applied to continuous distributions as well. However, Equation 2 requires entropy measures. While some entropy measures (such as the minimum relative entropy) are usually very hard to calculate, calculating the entropy of a distribution is feasible for both discrete and continuous cases. Hence, we believe continuous probability distributions may be used, if necessary. Furthermore, discrete probability distributions already provide more information than that provided by the ranges of opinion values of existing methods.

The first is an aggregation of objective opinions only, while the second is an aggregation of both objective and subjective opinions resulting in a final subjective measure. Hence, like individual opinions, group opinion is defined as a probability distribution that represents the probability that entity $\alpha$ is $e_i$ at time $t$ (or has the reputation of being $e_i$ at time $t$).

But why do we distinguish between these two different group opinions? Consider reputation in the field of football. Teams may play against each other. The results of these games may be viewed as having direct opinions being formed by one team about the other. For example, Barcelona winning Real Madrid 6–2 may be interpreted as Barcelona forming an opinion about Real Madrid being very weak and Real Madrid forming an opinion about Barcelona being very strong. This is one example of direct opinions in the field of football. Now consider that Real Madrid is either ranked high by some magazine or starts recruiting highly reputable players (at least higher than what they already have). In such a case, we might agree that this should increase the overall expectation of the team's performance, which could be viewed as increasing group opinion. Nevertheless, we believe such opinions are subjective, and their reliability cannot be matched to that of the objective group opinion (such as having Barcelona winning every single game for the last couple of years). Hence, we find differentiating between objective and subjective group opinion to be crucial.

Furthermore, we say subjective group opinion should lose its value with time, and move towards the objective one. For example, if Real Madrid kept on recruiting highly reputable players but failed to actually win their games, then the final reputation measure should always move towards the objective group opinion, i.e. the results of their games. Hence, we say, although subjective measures are important to describe the current group opinion, a purely objective measure is also needed. With time, and with the lack of new information, subjective group opinions should move towards objective ones. This is the notion of decay: everything loses its value with time. Similarly, objective group opinion would also decay towards the flat probability distribution (the distribution describing the state of complete ignorance), although at a presumably much slower rate.

The following section illustrates how these different measures may be calculated, highlights the links between them, and elaborates on the notion of decay.

## 3 The Proposed Model

This section focuses on the computation of group opinions, both objective ($\mathbb{D}$) and subjective ($\mathbb{P}$), in Sections 3.1 and 3.2, respectively. Reputation is then defined by Section 3.3.

### 3.1 The Default Opinion $\mathbb{D}$

We say the default opinion of an entity $\alpha$ is the group's opinion about $\alpha$ that is based on objective opinions only. The group's opinion is calculated by considering all the objective opinions expressed in the past, taking into account the certainty of each of these opinions.

**Assessing an objective opinion.** Assume that $\beta$ at time $t$ gives the following opinion about $\alpha$: $o^t_\beta(\alpha) = \{e_1/v_1, \dots, e_n/v_n\}$. We need to consider how much *value* this opinion has, based on how reliable is $\beta$ in giving opinions about $\alpha$.

In this model, we will consider that the overall reliability of any opinion is the reputation value of the entity expressing the opinion, which changes along time. This reputation value $\mathcal{R}$ is defined later on by Section 3.3. However, in this section, we use this reputation value (which we view as an indication of the reliability of the opinion $o^t_\beta(\alpha)$) to modify the opinion value. The basic idea is that the more reliable an opinion is, the closer the final value is to the original opinion, and the less reliable an opinion is, the closer the final value is to the flat (uniform) distribution $\mathbb{F}$ (where $\mathbb{F} = \frac{1}{|E|}$). Thus, we define the distribution representing $\beta$'s final view about $\alpha$ at time $t$ as follows:

$$\mathbb{O}^t_\beta(\alpha) = \mathcal{R}^t_\beta \times o^t_\beta(\alpha) + (1 - \mathcal{R}^t_\beta) \times \mathbb{F} \tag{1}$$

**The certainty of an opinion.** The group's opinion is based on an aggregation of individual opinions. However, the certainty of each of these individual opinions is crucial. We say, the more uncertain an opinion is then the smaller its effect on the final group opinion is. The maximum uncertainty is defined in terms of the flat distribution $\mathbb{F}$. Hence, we define this certainty measure as follows:

$$\mathcal{I}(\mathbb{O}^t_\beta(\alpha)) = \mathcal{H}(\mathbb{O}^t_\beta(\alpha)) - \mathcal{H}(\mathbb{F}) \tag{2}$$

where $\mathcal{H}(\mathbb{X})$ represents the entropy of a probability distribution $\mathbb{X}$. In other words, the certainty of an opinion is essentially the difference in entropy between the opinion and the flat distribution.

**Calculating $\mathbb{D}$.** Again, we note that an entity can give opinions on another one at different moments in time. So let us define by $T_\beta(\alpha) \subseteq T$ the set of time points in which $\beta$ has given opinions about $\alpha$. The default group opinion $\mathbb{D}^t_\alpha$ about $\alpha$ at time $t$ is then calculated as follows:

$$\mathbb{D}^t_\alpha = \frac{\displaystyle\sum_{\beta \in G} \sum_{t' \in T_\beta(\alpha)} \mathbb{O}^{t' \to t}_\beta(\alpha) \cdot \mathcal{I}(\mathbb{O}^{t' \to t}_\beta(\alpha))}{\displaystyle\sum_{\beta \in G} \sum_{t' \in T_\beta(\alpha)} \mathcal{I}(\mathbb{O}^{t \to t'}_\beta(\alpha))} \tag{3}$$

where, $\mathbb{O}^{t' \to t}_\beta$ represents the decayed value of $\mathbb{O}^{t'}_\beta$, and is discussed shortly.

This equation essentially states that the default group opinion is an aggregation of all $\mathbb{O}^{t' \to t}_\beta(\alpha)$ that represent the view of every entity $\beta$ that has formed an opinion of entity $\alpha$ at time $t$. However, different views are given different weights, depending on the certainty $\mathcal{I}(\mathbb{O}^{t' \to t}_\beta(\alpha))$ of these views.

**Initialising $\mathbb{D}$.** When an entity $\alpha$ is first introduced or created at time $t$, there is no information what so ever about this entity yet. Hence, its initial probability distribution is the flat distribution $\mathbb{F}$ that accounts for the maximum ignorance (i.e. the maximum entropy): $\mathbb{D}^t_\alpha(e_i) = \mathbb{F}(e_i) = \frac{1}{|E|}$. Along time, and as objective direct opinions are formed, this probability gets updated following Equation 3.

**Decaying $\mathbb{D}$ (and $\mathbb{O}$).** Like any other type of information, the default group opinion is expected to lose its value with time. For example, assume that a given player has played a lot of games and gained a high default opinion; however, for a very long time, this player has never played again. What can one say about the player's default opinion at the present time? Naturally, its glorious history does not necessarily mean that the player still has those old skills. Hence, we say that with time, $\mathbb{D}$ loses its value (very) slowly by decaying towards the flat probability distribution $\mathbb{F}$ according to the following equation:

$$\mathbb{D}^{t' \to t}_\alpha = \Lambda(\mathbb{F}, \mathbb{D}^{t'}_\alpha) \tag{4}$$

where $\Lambda$ is the *decay function* satisfying the property that $\lim_{t\to\infty} \mathbb{D}^t_\alpha = \mathbb{F}$. In other words, $\Lambda$ is a function that makes $\mathbb{D}^t_\alpha$ converge to $\mathbb{F}$ with time. One possible definition for $\Lambda$ could be: $\mathbb{D}^{t'\to t} = (\mathbb{D}^t - \mathbb{F})\nu^{\Delta_t} + \mathbb{F}$, where $\nu \in [0,1]$ is the decay rate, and $\Delta_t = 1 + (t - t')/\kappa$, where $\kappa$ determines the pace of decay.

Single opinions are pieces of information and as such they also decay along time. $\mathbb{O}^{t'\to t}_\beta$, which represents the decayed value of opinion $\mathbb{O}^{t'}_\beta$ at time $t$, is then similarly defined:

$$\mathbb{O}^{t'\to t}_\beta = \Lambda(\mathbb{F}, \mathbb{O}^{t'}_\beta) \tag{5}$$

### 3.2 The Inferred Opinion $\mathbb{P}$

While the default opinion $\mathbb{D}^t_\alpha$ represents the *objective* direct opinions of group members, the inferred opinion $\mathbb{P}^t_\alpha$ represents the final subjective opinion which is influenced by: objective direct opinions and subjective (both direct and propagated) opinions.

**Calculating $\mathbb{P}$.** How $\mathbb{P}$ is calculated differs with the different types of opinions triggering this calculation. The different cases are presented below.

1. **Subjective opinions.** If an entity is influenced by subjective opinions (whether direct or not), then its $\mathbb{P}^t_\alpha$ value is calculated accordingly:

$$\mathbb{P}^t_\alpha = \zeta\, \mathbb{P}^{t'\to t}_\alpha + (1-\zeta)\, \mathcal{X} \tag{6}$$

   where $\zeta$ is generally based on the reliability of $\alpha$ and $\mathcal{X}$ describes the new subjective opinion. This equation implies that when $\alpha$ is highly reputable, the effect of $\mathcal{X}$ is minimal, and vice versa. The exact values of $\zeta$ and $\mathcal{X}$ are dependent on the type of the subjective opinion, which we outline below:

   (a) *Direct subjective opinions.* In this case, we say $\zeta = (\mathcal{R}^t_\alpha)^{\mathcal{R}^t_\beta}$ and $\mathcal{X} = o^t_\beta(\alpha)$. In other words, if an entity $\beta$ forms an opinion about an entity $\alpha$, then $\mathcal{X}$ takes the value of $\beta$'s new opinion $o^t_\beta(\alpha)$. $\zeta$ would mainly be based

on the reliability of $\alpha$, but is also influenced by the reliability of $\beta$ since different entities should have different strength in affecting $\alpha$. We note that $\mathcal{R} \in [0, 1]$, as illustrated by Section 3.3.
In some cases, however, $\beta$ may be a foreign entity to the structural graph. Examples of this case are when a paper wins a award, or a magazine ranks football players. We assume that it is hard to know the reputation of foreign sources and their effect on $\alpha$. In such cases, the default value is $\mathcal{R}^t_\beta = 1$. Alternatively, the user may be free to assign a different reliability measure to $\mathcal{R}^t_\beta \in [0, 1]$.

(b) *Influential opinions of related entities.* In this case, we say $\zeta = (\mathcal{R}^t_\alpha)^{f(d_\alpha)}$ and $\mathcal{X}$=$\mathbb{P}^t_\beta$, where $f(d_\alpha)$=$(\mathcal{R}_\beta$+$d_\alpha$−$1)/d_\alpha$. In other words, if a neighbouring node $\beta$ (whether it was a parent or a child node) had its $\mathbb{P}^t_\beta$ value modified, then this should affect $\alpha$'s $\mathbb{P}^t_\alpha$ value. Again, the more reliable $\alpha$ is, then the smaller the effect of $\beta$ should be. Nevertheless, the effect of $\beta$ on $\alpha$ should also be influenced by the number of neighbouring nodes that $\alpha$ has (defined as $d_\alpha$, or the degree of $\alpha$). The larger this number, the smaller the effect of one neighbouring node is, and vice versa. We note that in this case, $d_\alpha \in [1, \infty]$. And the function $f(d_\alpha) = \mathcal{R}_\beta$ when $d_\alpha = 1$, and $\lim_{d_\alpha \to \infty} f(d_\alpha) = 1$.

2. **Objective opinions.** Objective opinions should have a stronger effect than subjective ones. In comparison with Equation 6, $\mathbb{P}^t_\alpha$ should now be calculated by giving more weight to the new objective opinion, as illustrated below:

$$\mathbb{P}^t_\alpha = \frac{\mathcal{R}^t_\alpha \; \mathbb{P}^{t' \to t}_\alpha + \mathcal{R}^t_\beta \; o^t_\beta(\alpha)}{\mathcal{R}^t_\alpha + \mathcal{R}^t_\beta} \tag{7}$$

Note that unlike Equation 6, even if $\alpha$ was fully reliable ($\mathcal{R}^t_\alpha = 1$), the new objective opinion of $\beta$ is still accounted for by taking into consideration the reliability of $\beta$ with respect to that of $\alpha$ (and vice versa).

**Initialising $\mathbb{P}$.** Similar to the default group opinion $\mathbb{D}$, we say $\mathbb{P}^t_\alpha(e_i) = \mathbb{F}(e_i)$, where $t$ is the time $\alpha$ is first introduced. Along time, this probability is updated according to the section above, either as opinions about $\alpha$ are formed by others, or as neighbouring entities have their $\mathbb{P}$s updated, influencing that of $\alpha$.

**Decaying $\mathbb{P}$.** The value of $\mathbb{P}$ is a subjective value, as it is influenced by subjective opinions. For example, the reputation of a team changes as it changes its team members, since opinions about new team members influence the opinion about the team. However, such information is subjective, and what really matters at the end is whether the team is actually capable of winning with this new group of team members or not. For this reason, we believe that with time, the subjective opinion $\mathbb{P}$ should decay at a reasonable rate towards a more stable and objective opinion: the default opinion $\mathbb{D}$. This is expressed by the following equation:

$$\mathbb{P}^{t' \to t}_\alpha = \Lambda(\mathbb{D}^t_\alpha, \mathbb{P}^{t'}_\alpha) \tag{8}$$

where $\Lambda$ is the *decay function* that has been introduced earlier by Equation 4.

### 3.3 Reputation and Reliability

As illustrated earlier, an essential point in evaluating the opinion of a given entity is how reliable ($\mathcal{R}^t_\beta$) it is. The idea behind the notion of reliability is very simple: an entity that is considered very good in a certain field is usually considered to be very good as well in assessing how others are in that field. This is based on the *ex cathedra* argument. An example of a current practice following the application of this argument is the selection of members of committees, advisory boards, etc.

But how is reputation calculated? Given an evaluation space $E$, it is easy to see what could be the 'best' opinion about someone: the 'ideal' distribution, or the 'target', which is defined as $\mathbb{T} = \{e_n \mapsto 1\}$. Given a 'target' distribution $\mathbb{T}$, the reputation of an entity $\beta$ may then be defined as the distance between the current default opinion $\mathbb{D}^t_\beta$ and the ideal distribution $\mathbb{T}$, as follows:

$$\mathcal{R}^t_\beta = 1 - \textsc{emd}(\mathbb{D}^t_\beta, \mathbb{T}) \tag{9}$$

where EMD is the earth movers distance that calculates the distance (whose range is $[1, 0]$) between two probability distributions [1]. [3][4] As time passes and opinions are formed, the reputation measure evolves along with the default opinion. We note that at any moment in time, the measure $\mathcal{R}^t_\beta$ can be used to rank the different entities.

## 4 The Algorithm

The proposed model of Section 3 illustrates how opinions may be inferred through the propagation (Equation 6) and aggregation (Equations 3, 6, and 7) of individual opinions in structural graphs. Algorithm 1 summarises this model.

We note that this algorithm runs locally for a given node $\alpha \in N$. The algorithm is invoked every time $\alpha$ receives a direct opinion $o^t_\beta(\alpha)$, or its neighbouring node $\beta$ updates its $\mathbb{P}$ value. We assume $\alpha$ saves all its computed $\mathbb{O}$ values (the value of the direct opinions it has received, following Equation 1) as well as its latest $\mathbb{P}$ and $\mathbb{D}$ values. The algorithm then proceeds by following the equations of the previous section in a straight forward manner.

## 5 Results

As illustrated by Figure 1, real life applications fall into different categories, based on whether they make use of objective opinions, subjective opinions, or both; or whether they make use of structural graphs or not. For example, Chess or Ping Pong are games with individual players, and the scores of the matches may be interpreted as objective opinions. The Diplomacy game is an example

[3] One important aspect to apply EMD is to determine what the distance between the terms in $E$ is. That is the matrix $D = \{d_{ij}\}_{i,j\in[1,n]}$. The distance is certainly domain dependent, and can possibly be learned.

[4] Naturally, other distance measurements may also be used.

**Algorithm 1** Updating node $\alpha$'s reputation $\mathcal{R}$ and inferred opinions $\mathbb{D}$ and $\mathbb{P}$

```
Require: $N$ to represent the nodes of the structural graph
Require: $G = \{\alpha, \beta, \dots\}$ a group of agents that may form opinions about nodes
Require: $E = \{e_1, \dots, e_n\}$ an evaluation space
Require: $t \in T$ to represent calendar time
Require: $o^t_\beta(\alpha)$ to represent the direct opinion that $\beta \in G$ holds about $\alpha \in N$
Require: $T_\beta(\alpha) \subseteq T$ to represent the set of time points in which $\beta$ has given opinions about $\alpha$
Require: $\mathbb{X}^t$ to represent the value of the probability distribution $\mathbb{X}$ at time $t$
Require: $\mathbb{X}^{t' \to t}$ to represent the decayed probability distribution $\mathbb{X}^{t'}$ at time $t$, following Equations 4, 5 and 8
Require: $get\_opinion(o^t_\beta(\alpha))$ to represent $\alpha$'s receipt of the direct opinion $o^t_\beta(\alpha)$
Require: $get\_neighbour\_update(\mathbb{P}^t_\beta)$ to represent $\alpha$'s receipt of the neighbouring node $\beta$'s updated $\mathbb{P}$ value
Require: $obj(o^t_\beta(\alpha))$ to represent that the direct opinion $o^t_\beta(\alpha)$ is an objective one
Require: $\mathcal{R}^t_\beta$ to represent $\beta$'s known reputation at time $t$
Require: $\mathcal{I}(\mathbb{O})$ to represent the certainty of the opinion $\mathbb{O}$, following Equation 2
Require: $d_\alpha$ to represent the degree of the node $\alpha$
Require: $f(d_\alpha, \mathcal{R}_\beta) = (\mathcal{R}_\beta + d_\alpha - 1)/d_\alpha$ which we simply refer to as $f(d_\alpha)$ when $\beta$ is obvious
Require: $\textsc{emd} : 2^{\mathbb{P}(E)} \times 2^{\mathbb{P}(E)} \to [0,1]$ which calculates the earth-mover distance between two probability distributions
  $\mathbb{F}(e_i) = \frac{1}{n}, \quad \forall\, e_i \in E$
  $\mathbb{T} = \{e_n \mapsto 1\}$
  when $get\_opinion(o^t_\beta(\alpha))$ do
    if $obj(o^t_\beta(\alpha))$ then
      $\mathbb{O}^t_\beta(\alpha) = \mathcal{R}^t_\beta \times o^t_\beta(\alpha) + (1 - \mathcal{R}^t_\beta) \times \mathbb{F}$
      $\mathbb{D}^t_\alpha = \dfrac{\sum_{\beta \in G} \sum_{t' \in T_\beta(\alpha)} \mathbb{O}^{t' \to t}_\beta(\alpha) \cdot \mathcal{I}(\mathbb{O}^{t' \to t}_\beta(\alpha))}{\sum_{\beta \in G} \sum_{t' \in T_\beta(\alpha)} \mathcal{I}(\mathbb{O}^{t \to t'}_\beta(\alpha))}$
      $\mathcal{R}^t_\alpha = 1 - \textsc{emd}(\mathbb{D}^t_\alpha, \mathbb{T})$
      $\mathbb{P}^t_\alpha = \frac{\mathcal{R}^t_\alpha \mathbb{P}^{t' \to t}_\alpha + \mathcal{R}^t_\beta o^t_\beta(\alpha)}{\mathcal{R}^t_\alpha + \mathcal{R}^t_\beta}$
    else
      if $\beta \in N$ then
        $\gamma = \mathcal{R}^t_\beta$
      else
        $\gamma = 1$
      end if
      $\mathcal{R}^t_\alpha = 1 - \textsc{emd}(\mathbb{D}^{t' \to t}_\alpha, \mathbb{T})$
      $\mathbb{P}^t_\alpha = (\mathcal{R}^t_\alpha)^\gamma \cdot \mathbb{P}^{t' \to t}_\alpha + (1 - (\mathcal{R}^t_\alpha)^\gamma) \cdot o^t_\beta(\alpha)$
    end if
  end when
  when $get\_neighbour\_update(\mathbb{P}^t_\beta)$ do
    $\mathcal{R}^t_\alpha = 1 - \textsc{emd}(\mathbb{D}^{t' \to t}_\alpha, \mathbb{T})$
    $\mathbb{P}^t_\alpha = (\mathcal{R}^t_\alpha)^{f(d_\alpha)} \cdot \mathbb{P}^{t' \to t}_\alpha + (1 - (\mathcal{R}^t_\alpha)^{f(d_\alpha)}) \cdot \mathbb{P}^t_\beta$
  end when
```

of individual players whose reputation is based on the subjective opinions of other team members. In Football, however, one may view a team as being composed of players and sometimes one player may play in different teams (based on the league), giving rise to the notion of a structural graph. Additionally, opinions about team players may sometime be subjective, such as being ranked by some magazine. Scientific publications may be viewed as an example that uses structural graphs (conference proceedings are composed of papers, papers are composed of sections, etc.), and opinions on scientific publications by other researchers in the field are subjective.

| | **uses structural graphs** | **doesn't use structural graphs** |
|---|---|---|
| **uses objective opinions** | Football | Chess<br>Ping Pong |
| **uses subjective opinions** | Football<br>Publications | Diplomacy |

**Fig. 1.** Categorised applications

We choose the Chess example for experimentation, because there exists an official ranking and predicting algorithm for Chess (ELO [2]) that we can compare to ours. Hence, for a given dataset that specifies the real outcome of games, we run the ELO algorithm and our proposed one to compute the reputation of players and predict the outcome of future games accordingly. We then compare the predicted outcome of each of the algorithms to the real one. Initially, we ran several experiments over real Chess data. However, we noticed that the performance of both the ELO mechanism and ours was similar. For instance, in one experiment, our algorithm performed 2.3% better than ELO. Looking at the results, it seemed that players in the same tournament are more or less of the same experience, and hence, reputation. For this reason, the final results of games were a little bit random, and hence, the performance of both ELO and this paper's proposed algorithm was similar.

We then moved on to simulated data. We created two players, $A$ and $B$, that played against each other over a number of years. $A$ was initially a 'bad' player and it lost around 80% of its games during the years 1992-1998. However, after 1998, $A$ stopped playing for a while, and it resumed playing in 2004. Its performance dramatically improved over the years 2004-2010. In general, our proposed algorithm performed better than ELO by 3.5%. We note that the results are still preliminary, as they simulate two players who play around 30 matches each. Figure 2 plots the distance between the real results and the predicted results of both ELO and our algorithm. The distance is measured using the earth mover's distance function, EMD; hence, the maximum distance possible is 1, and the minimum is 0. However, as illustrated by Figure 2, the main difference is highlighted in the year 2004, when the ELO algorithm performs very poorly compared to ours, since our algorithm's decay function allows a better prediction when behaviour changes with time.

As such, we conclude that our proposed algorithm is essentially useful in applications where the quality being assessed (behaviour of humans, performance of agents, quality of papers, etc.) could actually change with time.

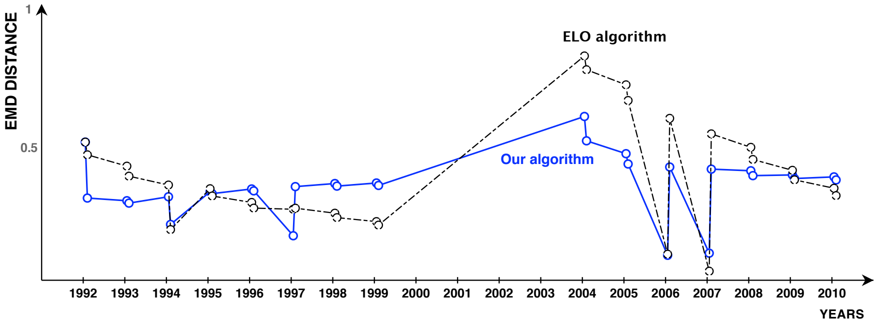


**Fig. 2.** The distance between predicted results and real results

## 6 Conclusion

This paper has proposed a model that allows agents to infer objective and subjective group opinion through the propagation and aggregation of opinions in structural graphs. It provides a clear distinction between objective and subjective opinions and their effect on group opinion. Additionally, the paper introduces the concept of opinion propagation between related entities.

In comparison with existing research, the research carried out by [3–5] studies the dynamics of opinion formation by focusing on the effect of social relations on how peoples' opinions may influence each other in a social network. The influence that one reviewer agent may have on another's subjective opinion is an interesting issue. Aggregation mechanisms, such as those presented by [6], may help in defining the appropriate aggregation method based on whether subjective opinions are dependent on each other or not. Repage [7], ReGreT [8], and SUNNY [9] provide mechanisms for computing the confidence in a reviewer based on the social relations. In this paper, we follow the *ex cathedra* argument which states that an agent's reputation could be used as an indication of its reliability in assessing others in its field. This fits perfectly in our equations that are concerned with objective opinions (Equations 1 and 7). However, again, when aggregating subjective opinions, social network analysis may be useful in contributing to the reliability of those opinions.

Concerning the propagation of opinions, we note that numerous research has addressed similar issues, such as [10–13]. PageRank [12] and Hits [13] calculate the relevance of web pages by analysing their position in the network and how they links to each other. Similarly, SARA [10] and CiteRank [11] present algorithms on how reputation may propagate based on who is citing whom. Their reputation propagates along citation links. This paper, on the other hand, focuses on the propagation of reputation along the structural links by focusing on the composition of entities and using the *part of* relation as an indication to the flow of opinions from one entity to another. Research work on ontology-based recommender systems, such as [14, 15], makes use of the clustering or classification of information and uses machine learning and data mining techniques for ranking and recommending entities. One may draw similarities between the taxonomies used by such systems and that of the structural graph of this document; although the propagation mechanism of this paper is unique in both its algorithm and semantics.

## Acknowledgement

This work has been supported by both the LiquidPublications project (`project.LiquidPub.org`), funded by the Seventh Framework Programme for Research of the European Commission under FET-Open grant number 213360, and the Agreement Technologies project (`www.agreement-technologies.org`), funded by CONSOLIDER CSD 2007-0022, INGENIO 2010.

## References

1. Peleg, S., Werman, M., Rom, H.: A unified approach to the change of resolution: Space and gray-level. IEEE Transactions on Pattern Analysis and Machine Intelligence **11**(7) (1989) 739–742
2. Elo, A.: The Rating Of Chess Players, Past & Present. 1st edn. Arco Pub. (1978)
3. Wu, F., Huberman, B.A.: Social structure and opinion formation. http://arxiv.org/abs/cond-mat/0407252 (July 2004)
4. Blondel, V.D., Guillaume, J.L., Hendrickx, J.M., Kerchove, C., Lambiotte, R.: Local leaders in random networks. Physical Review E **77**(3) (2008) 036114
5. Klimek, P., Lambiotte, R., Thurner, S.: Opinion formation in laggard societies. EPL (Europhysics Letters) **82**(2) (2008) 28008+
6. Sierra, C., Debenham, J.: Information-based reputation. In Paolucci, M., ed.: First International Conference on Reputation: Theory and Technology. (2009) 5–19
7. Sabater-Mir, J., Paolucci, M., Contexs, R.: Repage: REPutation and imAGE among limited autonomous partners. Journal of Artificial Societies and Social Simulation (JASSS'06) **9**(2) (2006)
8. Sabater, J., Sierra, C.: Social regret, a reputation model based on social relations. ACM, SIGecom Exchanges **3.1** (2002) 44–56
9. Kuter, U., Golbeck, J.: Sunny: A new algorithm for trust inference in social networks using probabilistic confidence models. In: Proceedings of the 22nd national conference on Artificial intelligence (AAAI'07), AAAI Press (2007) 1377–1382
10. Radicchi, F., Fortunato, S., Markines, B., Vespignani, A.: Diffusion of scientific credits and the ranking of scientists. http://arxiv.org/abs/0907.1050 (Sep 2009)
11. Walker, D., Xie, H., Yan, K.K., Maslov, S.: Ranking scientific publications using a simple model of network traffic. http://arxiv.org/abs/physics/0612122 (2006)
12. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab (1999)
13. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. Journal of the ACM **46**(5) (1999) 604–632
14. Bouza, A., Reif, G., Bernstein, A., Gall, H.: Semtree: Ontology-based decision tree algorithm for recommender systems. In: Proceedings of the Seventh International Semantic Web Conference (ISWC2008). (October 2008)
15. Ziegler, C.N., Schmidt-Thieme, L., Lausen, G.: Exploiting semantic product descriptions for recommender systems. In: Proceedings of the 2nd ACM SIGIR Semantic Web and Information Retrieval Workshop, Sheffield, UK (July 2004)

# Can You Do Me A Favor?

Keith Sullivan, Sean Luke, and Brian Hrolenok

Department of Computer Science, George Mason University
4400 University Drive, Fairfax VA 22030 USA
{ksulliv2, sean, bhroleno}@cs.gmu.edu

**Abstract.** Multiagent systems often require coordination among the agents to maximize system utility. Using the notion of favors, we propose a technique, *flexible reciprocal altruism*, which determines when one agent should grant a favor to another agent based on past interactions. The desired rate of altruism is controllable, and as a result the loss associated with granting unmatched favors is bounded and amortized over all past interactions. In flexible reciprocal altruism the desired acceptable loss is independent of the cost and value of the favors. Experiments show that our technique performs well with different cost/value tradeoffs, numbers of agents, and load.

## 1 Introduction

In a multi-agent system environment, the agents typically have limited resources (e.g., sensors, communications) that restrict the problems they can solve. If the agents coordinate, then the system can be made more effective. In particular, system performance generally improves if the agents can coordinate both locally and globally. Example applications include cooperative target observation [1], foraging [2], and peer-to-peer systems [3]. One useful approach to agent coordination is to consider the agents as providing *favors* to the other agents.

Most of the existing work on favors has naturally followed from some version of *reciprocal altruism:* one agent is willing to incur a cost *now* to provide a value to another agent in return for (hopefully) receiving value in the future from the other agent, who in turn will incur some cost. Existing work in reciprocal altruism in multiagent systems has focused to date on rational agents, trust, social laws, and reputation. We instead tackle the issue from the viewpoint of *amortized risk:* the degree to which an agent is willing to go out on a limb for another agent is restricted in such a way that if the second agent never reciprocates again, the first agent's loss, amortized over *all previous interactions* with the second agent, is bounded.

Why would an agent behave like this? We argue that in many multiagent systems tasks, a moderate number of agents are essentially working towards the common good, and have a reasonable expectation of similar behaviors from other agents. However, the agents do not yet know exactly the nature of the other agents. The other agents could be dishonest, of course, but often instead they may have a poor perception of their own abilities; or they may simply be swamped with work; or there may be noise in the system. In the latter situations, an optimistic view of one's neighbors is called for. We believe there is a broad array of possible real-world applications which yield this situation: for example, cooperative "swarm-style" unmanned vehicular foraging and observation; or peer-to-peer networks; or distributing tasks over heterogeneous grid computing systems.

Bounded amortized risk yields a very simple, almost simplistic, decision rule which we call *flexible reciprocal altruism*. In a nutshell: the sum total degree to which an agent $A$ is willing to provide help to another agent $B$ is simply a linear function of the help $A$ has received from $B$ in the past.

As discussed later, this is not Tit-for-Tat. One of the consequences of flexible reciprocal altruism is that as the agents cooperate back and forth, they begin to be willing to provide larger and larger favors to one another, and furthermore, we may control this rate of growth. In Tit-for-Tat the size of favors remains fixed, as is the version in [4].

## 2 Related Work

The social science and economic communities have devoted considerable attention to cooperation, in particular, the nature of altruism and why it evolved [5, 6]. Continuing the same line of reasoning, several researchers examined how reciprocity and altruism effect the development of social agents [7–10].

More closely related to our work, Trivers examined how reciprocal altruism evolved in nature [11]. Using Triver's work as motivation, Axelrod developed some of the original theory of cooperation in a game-theoretic setting [12]. He developed an evolutionary model based on the probability that two agents would interact again in the future, and showed how stable behavior arrises, when agents exhibit deterministic reciprocity towards each other. Sen extended this work to probabilistic reciprocity, using publicly available discount factors to encourage sharing of resources [4, 13–15]. That work, in various guises, generally bases the probability of agent $A$ assisting agent $B$ directly on the historical *difference* in cost and benefit to $A$ of interacting with $B$. This is related to our approach, which will instead increase the degree to which $A$ helps $B$ based on the *ratio* of historical cost and benefit; thus as $A$ and $B$ interact more, $A$ is willing to help $B$ to a larger and larger extent. We elaborate on a portion of this work [4] in Section 3. Hazard further broadened Sen's work by using private discount factors [16].

Altruism is also studied in cooperative game theory [17], which deals with coalition formation. Coalitions are groups of agents where the benefit received by an individual agent is higher when acting with the group than when acting alone. Shapley showed that under certain conditions the core of such a game is non-empty [18]. In other words, there exists a coalition structure such that no agent has incentive to change (similar to the idea of Nash equilibria). Fuzzy coalition theory seeks a middle ground between coalitions and self-interested agents [19, 20]. Economists have studied reciprocity and favors using cooperative game theory and fuzzy coalition theory, focusing on the economy that develops when favors are associated with a cost and a benefit [21–24]. A similar approach was used to improve wireless sensor networks between two organizations [25].

Closely related to reciprocity and cooperative game theory is the notion of trust and its extension in the form of reputation. Since interactions form the basis of a multi-agent system, the multi-agent community has devoted considerable effort towards understanding trust in large scale systems [26, 27]. Hand in hand with trust is the idea of reputation [28, 29]. While trust and reputation are significant research areas within multi-agent systems, our flexible reciprocal altruism does not use either mechanism per se; instead it relies on historical pairwise interactions.

## 3 Model Description

In flexible reciprocal altruism, an agent grants favors by considering the degree to which it has had favorable interactions with the grantee agent in the past. If the grantee agent has provided many significant favors in the past to the grantor, the grantor is willing to offer the grantee more unmatched favors in the future. Imagine that agent $i$ is determining whether to grant agent $j$ a favor. Let $v_{ji}$ be the sum total *value* to agent $i$ of the favors which agent $j$ has granted agent $i$ up to this point. Let $c_{ij}$ be the sum total *cost* to agent $i$ of the favors which agent $i$ has granted agent $j$ up to this point. Agent $i$ will grant the proposed *favor* if:

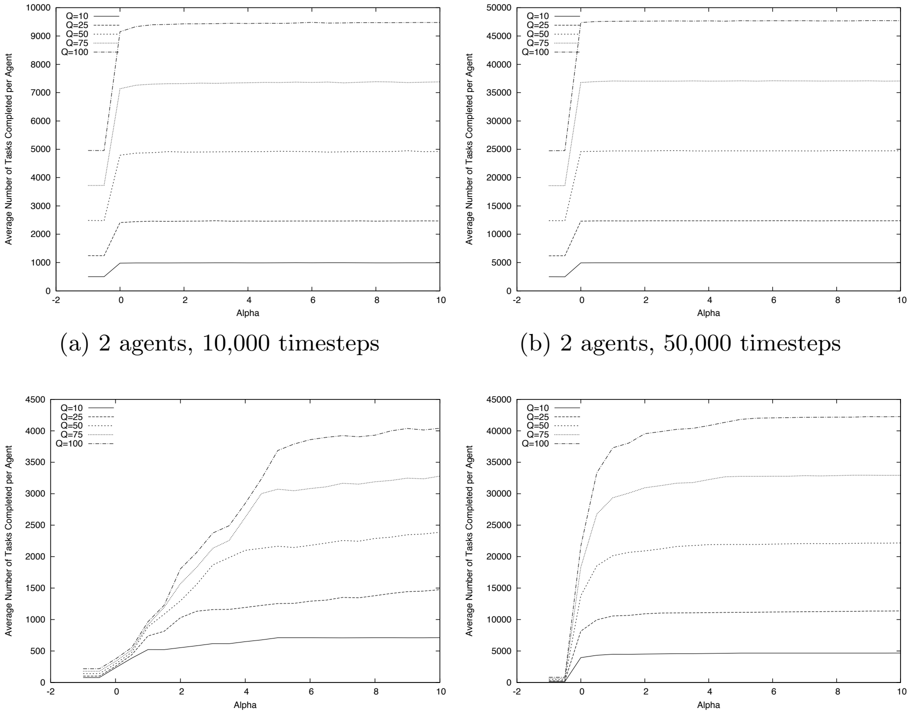
$$c_{ij} - v_{ji} + \mathrm{cost}(\mathit{favor}) \leq \alpha v_{ji} + \beta.$$

The idea here is that the degree to which agent $i$ will go out on a limb for agent $j$ (that is, $c_{ij} + \mathrm{cost}(\mathit{favor}) - v_{ji}$) is no more than some constant ($\alpha$) times how well agent $j$ has treated agent $i$ in the past (that is, $v_{ji}$). $\alpha$ is essentially a measure of risk tolerance: highly altruistic agents may have a high value of $\alpha$, whereas risk-averse agents will have a low value. An initialization constant, $\beta > 0$, determines the initial level of altruism, i.e., how many favors should an agent grant initially before requiring the other agent to reciprocate. Without $\beta$, agents would never grant favors to grantees with which they have had no history (that is, when $v_{ji} = 0$). Rearranging, we get:

$$\beta + (1 + \alpha)v_{ji} \geq c_{ij} + \mathrm{cost}(\mathit{favor}). \tag{1}$$

This is our decision equation: a favor will be granted so long as it does not increase $c_{ij}$ to greater than $\beta + (1 + \alpha)v_{ji}$. This decision equation is not based on rational agents per se: rather, it is based on amortized altruism. Imagine that agent $i$ and $j$ have had a history of interactions, and then agent $j$ cheats $i$ and walks away. To agent $i$ this is acceptable if the *amortized ratio of cost to benefit over all interactions with* $j$ does not exceed some amount. Clearly from the previous equation, $c_{ij} = O(v_{ji})$, and (rearranging yet again) the augmented cost to benefit ratio is:

$$\frac{c_{ij} + \mathrm{cost}(\mathit{favor}) - \beta}{v_{ji}} \leq 1 + \alpha. \tag{2}$$



(a) 2 agents, 10,000 timesteps

(b) 2 agents, 50,000 timesteps

(c) 64 agents, 10,000 timesteps

(d) 64 agents, 50,000 timesteps

**Fig. 1.** Average number of completed tasks for 2 and 64 agents and various task queue lengths ($Q$).

As the agents gain a transaction history and $c_{ij}$ and $v_{ji}$ both grow, $\beta$ becomes inconsequential and the ratio of cost to benefit approaches $1 + \alpha$.

*Tit-for-Tat and the Reciprocal Norm* It's worthwhile to compare this rule to others. Using our rule, if an agent $A$ is being asked by another agent $B$ for a string of favors, it will provide them up to its level of risk tolerance ($\alpha$). Initially the allowed strings will be small, but ultimately they may be very large if needed. Because we can tune $\alpha$, the rate of growth of allowed strings is not dependent on the cost and value of favors.

Now consider a simple variation of Tit-for-Tat which disregards the relative costs and values of the tasks to the agents. Here, agent $A$ will only grant $B$ a favor if the number of favors $A$ has granted $B$ in the past is less than or equal to the number $B$ has granted $A$. If $B$ needs a string of favors in a row, $A$ will provide only enough of them to achieve parity plus one favor, and no more

We take liberties with the simple variation of Tit-for-Tat to consider costs and values in the tasks. In this "Expanded" version, an agent $A$ will grant $B$ a favor only if the *cost* to $A$ of the favors $A$ has granted $B$ in the past is less than or equal to the *value* to $A$ of the favors $A$ has received from $B$. Assuming that all tasks have the same cost and the same value, this is equivalent to Equation 1 with $\alpha = 0$. In this model the strings of favors agents offer one another will grow without bound if the cost of a task is less than the value of the task. However this growth rate is entirely determined by the ratio of cost to value, since $\alpha$ is disregarded. Furthermore, if the cost and value are the same, then this model reduces to the simple Tit-for-Tat scenario earlier.

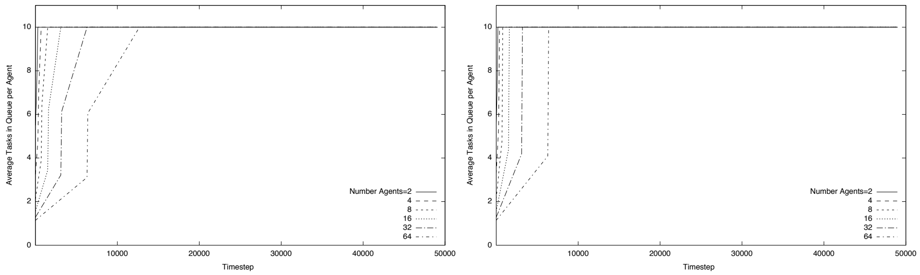
A more elaborate system described in [4] permits consideration of various task types with different costs and values, expectation of future distribution of tasks by type, and a probability model of other agents providing future favors. The authors suggest a reduced version in which both the future distribution of tasks and the future favor probability model are estimated using the results from the immediately previous time step (we may generalize this to some window of previous time steps). If we employ a single task type with a given cost and value, as is done in the experiments in this paper, the model may be simplified to $Pr_{A,B} \overset{?}{<} Pr_{B,A}$, where $Pr_{i,j}$ is the recent probability that, when asked, $i$ has given $j$ a favor. We believe this is a manifestation of the so-called *reciprocal norm*, that is, $A$ gives $B$ favors because he feels indebted to $B$. We would augment this to $Pr_{A,B} \overset{?}{\leq} Pr_{B,A}$ to enable granting when the two are evenly matched. An approach along these lines, like Tit-for-Tat, fixes the length of favor strings if the two agents are evenly matched.

We note that our method is also in some sense more robust to noise once the agents have warmed to one another. Imagine if agent $A$ believes he has given agent $B$ a favor or two, but $B$ did not receive them due to noise. In the three systems described above, ultimately the situation may arise where each agent believes the other owes him, and refuses further interactions. In our approach, depending on how many interactions $A$ and $B$ have had with one another, this situation can only arise after a large number of noise-caused failures.

*Free Riders* Because of its optimism, our approach is susceptible to free-riding agents ("leeches") in two ways. First, if an agent $A$ meets an unknown agent $B$ for the first time, $A$ will always agree to offer $B$ a small initial favor no greater than $\beta$. Were there an infinite number of agents, $B$ could wander about forever, getting one favor from each agent. Second, a smart agent $B$ could ratchet up the favors he steals from $A$ by building up a transaction history with $A$, then asking for a large number of favors, then walking away.

It's important to note that, from the perspective of $A$, the first situation is in some sense worse. Consider the case where the cost and value of favors are both 1. For $B$ to prime $A$ to give out an unmatched string of $N$ favors, $B$ would have had to provide $A$ with some $\frac{N-1}{\alpha}$ favors in turn (and vice versa). But if some $N$ rogue friends of $B$'s each asked $A$ for one favor, $A$ would offer it to all of them without any reciprocation.

In a finite-sized group, free-riders of the first kind are capped automatically: they'll run out of people to cheat. Free-riders of the second kind can extract any single amount they wish from the altruistic agents, but to extract more they will need to give back more. The total amount they can cheat the system



(a) $\alpha = 0.0$, 50,000 timesteps (b) $\alpha = 1.0$, 50,000 timesteps

**Fig. 2.** Average queue length per agent. Maximum queue length was 10.

is unlimited, but the *ratio* of loss, with respect to the value they provided the system, is bounded. Importantly, *this ratio may be set by the altruistic agents:* if they are risk averse they may cut $\alpha$ clear to 0, in which their loss is never more than $\beta$ (here, 1). Or they can cut $\alpha$ clear even further, down to -1, at which point they never grant more than a single favor to a requester, and ignore him after that.

*Favor Brokers* Though our approach allows agents to warm to one another relatively quickly regardless of the ratio of cost to value, it also makes possible extensions which promise even faster warming. We are particularly interested in the notion of *favor brokers* which can act as a bridge between two agents who otherwise know little about one another. Consider two agents $A$ and $C$ who do not know one another well. $A$ needs a significant favor but $C$ ordinarily would not provide it because of limited transaction history between the two. However, broker $B$ has a significant history with both $A$ and $C$. $A$ asks $B$ for a favor, and $B$ agrees to broker it by in turn asking $C$ to perform the favor. $C$ agrees to do the favor for $B$. $A$ receives the value and $C$ the cost of the favor; but $C$ believes that $C$ has done $B$, not $A$, a favor ; and $A$ likewise believes that $B$, and not $C$, has done $A$ a favor. Note that neither $A$ nor $C$ have changed in opinion about one another.

The idea behind brokerage is to provide a special channel whereby a population may offer favors despite little prior history; or for two alien populations to offer favors to one another. Though this approach is related to notions of reputation and trust (for example [29]) it is *different* in an important way: $B$ is not introducing $A$ and $C$ to one another. At the end of the day, $A$ and $C$ have no additional transaction history. $B$ has not used its reputation to convince $C$ that $A$ is worthwhile; rather $B$ has assumed responsibility as a middle-man for the transaction.

Though it would be interesting for future study, we do not at this time permit transitivity in brokerage.

## 4 Experiments

Our experimental problem is an abstraction of common basic factory-floor models. Each agent has a queue of a certain *length* (how many tasks it can hold), and each timestep it removes a single task from its queue, if there is any, and performs it. As time passes, new tasks stochastically arrive for each agent, sometimes many at a time. If the agent's queue is not yet full, it will put the tasks in its own queue. Otherwise the agent will ask others for favors: to put the tasks in their queues instead. If an agent cannot curry a favor, and his queue is full, the task will be dropped on the floor and be lost.

When an agent places a task into his queue (to perform it in the future) he immediately incurs a *cost*, but the task's owner receives a *value*. Costs are less than or equal to values: and in our experiments all tasks have the same cost and the same value. Thus granting a favor costs the grantor in two ways. First, when a grantor performs a grantee's task, the grantor incurs the cost but the grantee receives the value. Second, by accepting tasks, the grantor runs the risk of filling his own queue such that newly arriving tasks of his own are dropped on the floor.

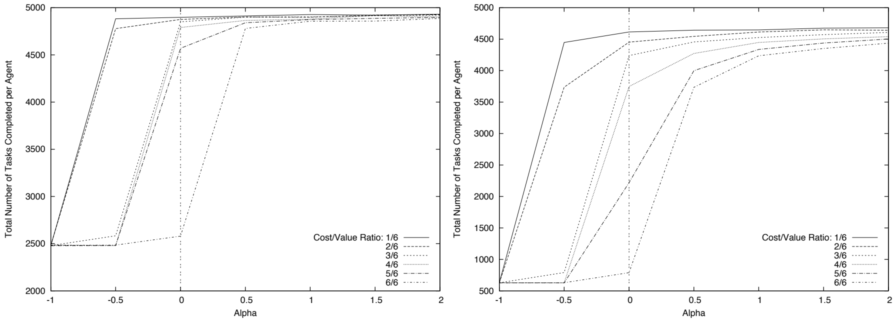
Tasks arrive following a Poisson distribution with mean of 100 timesteps, and normally round robin selection determines which agent receives the tasks. When asking for a favor, agent $A$ asks a random agent $B$, and if $B$ does not grant the favor, then $A$ chooses another random agent to ask, then another, and so on. If ultimately no agent grants $A$'s favor request, then the task is lost. $\alpha$ usually ranged from -1 to 10 in steps of 0.5. Runtime was set to either 10,000 or 50,000 timesteps. All experiments were done in the MASON simulator [30] and the results were averaged over 50 independent runs. Except in Experiment 3, task value was set to 3 and task cost to 2.

Our primary statistic was number of tasks performed per agent in the allotted time. We chose this, rather than total value minus total cost, because it is insensitive to the particular task costs and values chosen (in particular, if task cost and value are the same, then total value minus total cost will always be zero). Another statistic was the average number of tasks completed per agent in a given timeframe.

*Experiment 1: Task Load, Number of Agents, and Degree of Altruism* We first set out to examine the dynamics of the system: what happens with increasing load (shorter queue lengths), more agents, and varying amounts of altruism ($\alpha$). Figure 1 shows the results. In the left column we show the results for 10,000 timesteps, which emphasizes the warm-up period until the agents were offering large favors to one another; and in the right column we show 50,000 timesteps, which is closer to the steady state of the system.

As would be obvious, smaller queue lengths resulted in more tasks being dropped on the floor, and thus lower numbers of tasks completed per agent.

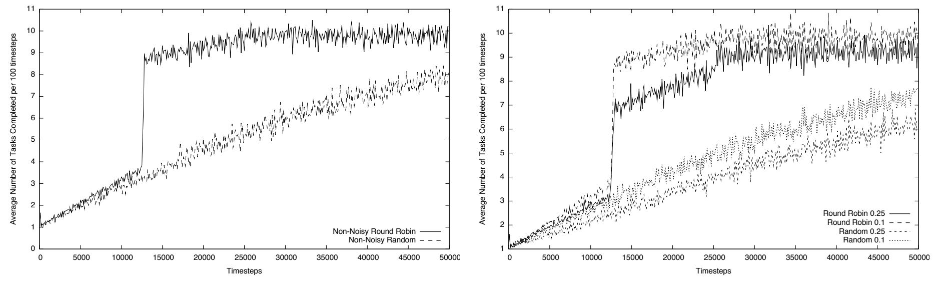
Figure 1 also demonstrates that the more altruistic agents (with larger values of $\alpha$) are able to warm up to one another more rapidly. Agents with $\alpha = -1$ (approximately greedy agents who only provide just one favor per grantee for the



(a) 2 agents, queue length 50

(b) 8 agents, queue length 50

**Fig. 3.** Total number of tasks per agent (10,000 timesteps), with different ratios of cost/value for each task. "Expanded Tit-for-Tat" ($\alpha = 0$) is marked with a vertical line to distinguish it.



(a) No Noise

(b) Different Noise Levels

**Fig. 4.** Average number of tasks completed per agent per 100 timesteps for round-robin and random task distribution, $\alpha = 1.0, \beta = 2.0$, queue length = 10, 125 agents

entire duration of the run) cannot improve. "Expanded Tit-for-Tat" style agents, with $\alpha = 0$, are also able to warm to one another because their decisions are based on the ratio of value to cost rather than on the specific number of favors granted. But for agents with larger $\alpha$ values, the rate of warming increases even more, resulting in more tasks competed per agent.

Last, but most importantly, it would seem intuitive that having more agents would allow for more favor opportunities, and this in turn should improve the average number of tasks performed. But in fact this is not the case: as he spreads his favors over more agents at random, a grantee takes longer to warm up to any given grantor. As a result, the total number of tasks performed is reduced until the agents have adequately warmed to one another. Ultimately, however, the steady state results converge to the same values regardless of the number of agents.

*Experiment 2: Queue Saturation* To verify that larger numbers of agents were in fact taking longer to warm up, we examined how full agents' queues were during the task. Figure 2 shows the number of tasks waiting in a queue, per agent, for $\alpha = 1$ (other $\alpha$ values were similar). The maximum queue length was 10. Note that as the number of agents increases, there are fewer interactions, and so the agents are idle until much longer in the run; however, at the steady state, eventually all the queues are filled.

*Experiment 3: Tit-for-Tat* Experiment 1 showed that "Expanded Tit-for-Tat" ($\alpha = 0$) would in fact warm to larger and larger favors. However, this is mostly due to the cost/value ratio involved. By default we had set this ratio to $\frac{2}{3}$. But if the cost and value of a task were the same, then Tit-for-Tat can do scarcely better than approximately greedy agents ($\alpha = -1$). This is shown in Figure 3. As the cost/value ratio approaches 1, Tit-for-Tat starts performing poorly. In fact, the more agents involved, the worse Tit-for-Tat performs, since each agent must make up favors over more agents without a warm-up mechanism. Larger values of $\alpha$ (that is, more altruistic agents) are much less sensitive to this ratio.

*Experiment 4: Task Allocation and Noise* Next we studied how changing which agent receives the tasks effects performance. We examined two methods for choosing which agent receives tasks: round robin selection (our previous default) and random selection. In this experiment, we also added noise to the system by increasing the probability that the grantor does not receive any value. In other words, the grantee will think it granted a favor while the grantor will think the favor was not granted.

We set the number of agents to 125, $\alpha = 1.0$, $\beta = 2$, and the queue length to 10. For the noisy experiments, we set the noise level at 0.1 and 0.25 (a noise level of 0.1 meant that 10% of the time, the grantor would not receive any value).

Figure 4(a) shows that changing how agents receive tasks alters performance. In the round-robin case, after all the agents have received the tasks once (around timestep 12,500), all the agents will then grant one another a favor; thus the spike in the graph. If instead we chose agents at random to receive tasks, the agents require additional time to build a history, thus resulting in poorer performance. Figure 4(b) shows how noise effects system performance. As expected, increasing the noise level caused performance to drop. However, our model formulation overcomes the noise, although it requires more time to do so.

*Experiment 5: Initially Altruistic Societies* This experiment studied one possible method to decrease the warmup period. In flexible reciprocal altruism, $\beta$ controls how *initially* altruistic the society is: as $\beta$ increases, agents are willing to grant more initial favors before expecting the other agent to reciprocate. In other words, increasing $\beta$ decreases the warmup period.

Figure 5 shows how the length of the warmup period changes with $\beta$, given $\alpha = 1.0$, maximum queue length of 10, and 125 agents. We ran the experiment for 50,000 timesteps but truncated the figure to 20,000 timesteps to emphasize the warmup period; after 20,000 timesteps all the curves are statistically the

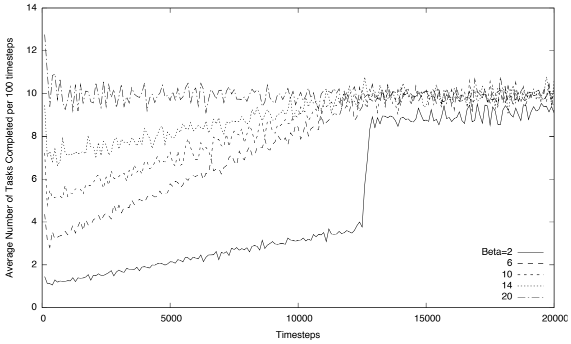


**Fig. 5.** Average number of tasks completed for different levels of initial altruism. $\alpha = 1.0$, queue length = 10, and 125 agents.

same. Note that for a very altruistic society (high $\beta$) the average throughput is maximal from the beginning.

*Experiment 6: Favor Brokers* Finally, we studied how introducing favor brokers influences performance. Recall that a *broker* is an intermediary who can ask a favor of one agent on behalf of another, despite possible distrust between the two. In a society with low altruism, brokers (and their trusted status) offer a potential way to increase interactions amongst the agents, and thus decrease the warmup period. Figure 6 shows the average number of tasks completed per 100 timesteps per agent, for $\alpha = 1.0$, $\beta = 2.0$, a maximum queue size of 10 and 50 agents. Increasing the percentage of brokers in the population increases the number of interactions between agents, thus causing quicker convergence even in a society that is not very altruistic. Changing $\alpha$ and the number of agents does not significantly alter the results.

This experiment, along with Experiment 5, suggests that either increasing the global level of altruism, or increasing the number of brokers within the society, will decrease the warmup period. While both techniques could be combined, the problem domains where such an arrangement is sensible may be limited.

## 5 Conclusions and Future Work

We have presented a simple formula which enables cooperating agents to warm to one another, offering larger and larger favors based on past experience. Though at any time an agent can cheat another agent and walk away, the second agent's amortized loss is bounded. This optimistic approach, which we have termed flexible reciprocal altruism, allows agents to optimize the collective performance of the system, and is particularly apropos to environments with moderate numbers of agents, significant and highly variable task load, and moderate noise. Our technique adapts faster and more flexibly than Tit-for-Tat and similar tech-

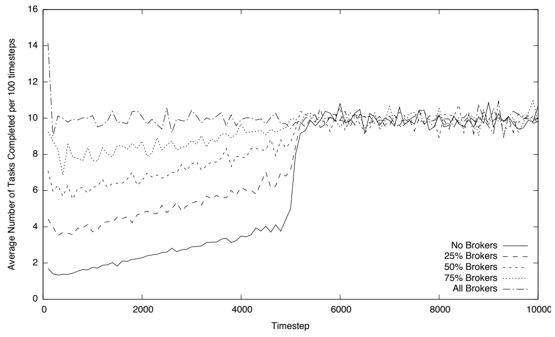


**Fig. 6.** Average number of tasks completed per 100 timesteps for varying percentage of brokers in the population. $\alpha = 1, \beta = 2$, queue length = 10, and 50 agents.

niques, and can handle situations they cannot: such as when the cost and value of a task are the same.

For future work: our existing approach lacks a time-discounting procedure: agents have perfect memory of all previous favor transactions. We have not yet examined the effects of different ways of doing this: for example, what would be the result of having longer memories for costs than for values?

We may also examine possible approaches to transitivity in favor brokerage, and using brokers to introduce entire disjoint populations to one another. Last, we have not experimented with per-agent values of $\alpha$, or different $\alpha$ values on a pairwise basis. It might also be helpful to consider dynamic $\alpha$ values, producing nonlinear functions or ones designed to cut off freeloaders more effectively.

*Acknowledgments* This work was supported in part by NSF grant 0916870.

## References

1. Luke, S., Sullivan, K., Balan, G., Panait, L.: Tunably decentralized algorithms for cooperative target observation. In: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, Utrecht Netherlands (July 2005)
2. Gheorghe, M., Holcomb, M., Kefalas, P.: Computational models of collective foraging. Biosystems **61**(2–3) (July 2001) 133–141
3. Camorlinga, S., Barker, K., Anderson, J.: Multiagent systems for resource allocation in peer-to-peer systems. In: Proceedings of International Symposium on Information Communication and Technologies. (2004) 1–6
4. Saha, S., Sen, S., Dutta, P.S.: Helping based on future expectations. In: Proceedings of Autonomous Agents and Multiagent Systems. (2003)
5. Dugatkin, L.A., Wilson, D.S., III, L.F., Wilkens, R.T.: Altruism, tit for tat and 'outlaw' genes. Evolutionary Ecology **8**(4) (1994)
6. Lehmann, L.: The evolution of cooperation and altruism–a general framework and a classification of models. Evolutionary Biology **19**(5) (2006)
7. Briggs, W., Cook, D.: Flexible social laws. In: Proceedings 14th International Joint Conference on Artificial Intelligence. (1995) 688–693

8. Grimaldo, F., Lozano, M., Barber, F.: A multiagent framework to animate socially intelligent agents. In: Innovations in Hybrid Intelligent Systems. Springer (2008)
9. Lerman, K., Shehory, O.: Coalition formation for large-scale electronic markets. In: Proceedings of Fourth International Conference on Multi-Agent Systems. (2000)
10. Matsubayashi, K., Tokoro, M.: A collaboration mechanism on positive interactions in multi-agent environments. In: Proceedings of International Joint Conference on Artificial Intelligence. (1993)
11. Trivers, R.L.: The evolution of reciprocal altruism. The Quarterly Review of Biology **46**(1) (1971) 35–57
12. Alexrod, R.: The Evolution of Cooperation. Basic Books (1984)
13. Saha, S., Sen, S.: Reciprocal negotiation over shared resources in agent societies. In: Proceedings of Autonomous Agents and Multiagent Systems. (2007)
14. Sen, S.: Reciprocity: A foundational principle for promoting cooperative behavior among self-interested agents. In: Proceedings of the Second International Conference on Multiagent Systems. (1996) 332–329
15. Sen, S.: Believing others: Pros and cons. Artificial Intelligence (2002)
16. Hazard, C.J.: ¿Por favor? Favor reciprocation when agents have private discounting. In: Proceedings of the 2008 AAAI Workshop on Coordination, Organizations, Institutions and Norms. (July 2008) 9–16
17. von Neumann, J., Morgenstern, O.: Theory of Games and Economic Behavior. Princeton University Press (1953)
18. Shapley, L.S.: Cores of convex games. International Journal of Game Theory **1**(1) (1971) 11–26
19. Branzei, R., Dimitrov, D., Tijs, S.: Models in Cooperative Game Theory: Crisp, Fuzzy, and Multi-Choice Games. Springer (2005)
20. Mareš, M.: Fuzzy Cooperative Games: Cooperation with Vague Expectations. Physica-Verlag (2001)
21. Abdulkadiroğlu, A., Bagwell, K.: Trust, reciprocity and favors in cooperative relationships. In: Proceedings of University of Maryland, Department of Economics, Workshop in Industrial Organization and Microeconomic Theory. (January 2005)
22. Fung, K.K.: Doing well by doing good: A market for favors. Cato Journal **15**(1) (1995)
23. Hauser, C., Hopenhayn, H.: Trading favors: Optimal exchange and forgiveness. In: Society for Economic Dynamics Annual Meetings. (2004)
24. Möbius, M.M.: Trading favors. Manuscript (May 2001)
25. Miller, D.A., Tilak, S., Fountain, T.: "Token" equilibria in sensor networks with multiple sponsors. In: Proceedings of International Conference on Collaborative Computing: Networking, Applications and Worksharing. (2005)
26. Ramchurn, S.D., Huynh, D., Jennings, N.R.: Trust in multi-agent systems. The Knowledge Engineering Review **19**(1) (2004) 1–25
27. Wang, Y., Singh, M.P.: Formal trust model for multiagent systems. In: Proceedings of International Joint Conference on Artificial Intelligence. (2007)
28. Pujol, J.M., Sangüesa, R., Delgado, J.: Extracting reputation in multi agent systems by means of social network topology. In: Proceedings of Autonomous Agents and Multiagent Systems. (2002) 467–474
29. Sabater, J., Sierra, C.: Reputation and social network analysis in multi-agent systems. In: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems, New York, NY, USA, ACM (2002) 475–482
30. Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., Balan, G.: MASON: A multiagent simulation environment. Simulation **81**(7) (July 2005) 517–527