Trust Alignment: A Sine Qua Non of Open Multi-agent Systems

Andrew Koster[1,2], Jordi Sabater-Mir[1], and Marco Schorlemmer[1,2]

[1] IIIA - CSIC
[2] Universitat Autònoma de Barcelona
Bellaterra, Spain

Abstract. In open multi-agent systems trust is necessary to improve cooperation by enabling agents to choose good partners. Most trust models work by taking, in addition to direct experiences, other agents' communicated evaluations into account. However, in an open multi-agent system other agents may use different trust models and as such the evaluations they communicate are based on different principles. This article shows that trust alignment is a crucial tool in this communication. Furthermore we show that trust alignment improves significantly if the description of the evidence, upon which a trust evaluation is based, is taken into account.

1 Introduction

A prerequisite for cooperation is that an agent may reasonably expect this cooperation to succeed. The cooperating agents need to know that their interaction partner will perform the action it agreed to. In many systems this can be enforced by the architecture of the system, however in open systems in which the individual agents maintain their autonomy, such as e-Commerce or smart electricity grids, this type of guarantee is not available and agents may be capable of cheating, lying or performing other unwanted behaviour. In such open multi-agent systems the agents need to choose selectively whom to cooperate with and trust is a fundamental tool for performing this selection.

Unfortunately, it is more complicated than equipping an agent with one of the available computational trust models [1] and expecting it to function in a social environment. Using trust as a method for picking successful cooperation partners relies not only on having a good trust model, but also on communication of trust evaluations with other agents [2]. This communication is far from straightforward, because trust is an inherently subjective concept [3]. In this paper we show that to communicate trust evaluations between agents some form of *trust alignment* is needed.

The subjectivity of trust can be seen in the following example, which also demonstrates why this is problematic for communication: consider an e-Commerce environment in which two agents buy the same bicycle via an online auction. One may evaluate the sale as very successful, because the bicycle was

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011, Part I, LNCS 7044, pp. 182–199, 2011.

cheap and in good condition. The other, however, puts more emphasis on delivery time and, since the seller delayed significantly before sending, it gives the seller a negative evaluation. Despite having identical interactions, the two agents differ significantly in their trust evaluations of the seller agent. If one of these agents had asked the other agent for advice regarding the seller, that advice would not have been *accurate* within the receiving agent's frame of reference, because the two agents support their trust evaluations with different aspects of the interaction. This problem extends to all domains in which open multi-agent systems may be applied. If trust evaluations – and other subjective opinions – are to be communicated accurately in such domains, a different set of tools is required than is used for the communication of facts. In [4] this is referred to as trust alignment and a couple of different methods for such alignment are suggested.

In this article we discuss the methods used to solve the problem and show, through experimentation, firstly that trust alignment is necessary for effective communication about trust, and secondly that, for truly effective alignment, the evidence on which a trust evaluation is based needs to be taken into account. This experimentation is detailed in Section 3 and the results are discussed in Section 4 before concluding the article in Section 5. The next section further introduces the problem of trust alignment and the proposed solutions to it.

2 Methods for Aligning Trust

Trust alignment is a method of dealing with the problem of interpreting another agent's trust evaluations, despite knowing that such evaluations are entirely subjective. As such it is classified as a problem of semiotic, or pragmatic, alignment [5]. While such problems are described in the field of semantic alignment, very little work has been done on finding solutions. Despite this, the field of semantic alignment provides a valuable framework [6] in which to define the problem of trust alignment. We can define trust alignment as the process of finding a translation of the other agent's trust evaluations, based on shared evidence. Its result is a method to translate other trust evaluations from the same agent, based on non-shared evidence. With evidence we mean an objective description of some artifacts in the environment, such as interactions the agents have participated in. Shared evidence is an objective description of an artifact which both agents have perceived, while non-shared evidence refers to artifacts which the receiving agent has not perceived. By using the shared evidence as a common ground, two agents can communicate their differing trust evaluations based on the same evidence and use these different evaluations of the same object as the starting point for finding a translation.

With this definition we can analyze the various processes which could serve to find such a translation. While many trust models have been proposed for computational agents in a multi-agent system [1], very few consider the interpretation of other agents' evaluations as being problematic. Of those that do, the majority are attempts at distinguishing between honest and dishonest agents.

Approaches such as those described by [7,8] attempt to find lying agents and discard all trust evaluations received from them. However, by discarding this information such methods run the risk of missing out on a lot of information; not because the communicating agent is dishonest, but because it has a different underlying trust model. Especially in an open multi-agent system it cannot be assumed that any agent with a differing opinion is being untruthful, although there may very well be such untruthful agents in the system. Detecting these is a separate and important problem, which such reputation filtering methods deal with, however these methods cannot properly be considered solutions to the problem of trust alignment.

By realizing trust alignment is first and foremost a problem of alignment, a number of common ontologies have been proposed to bridge the gap between different trust models [9,10]. 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 [11], but all these approaches are limited: they align the meaning of the concepts of trust, but not how an agent arrives at, or uses, a specific evaluation, and thus they do not deal with the fact that trust evaluations are subjective. To clarify this distinction we refer back to the example in the introduction: the agents disagree on *how* to evaluate a target, with one agent giving more importance to cost and quality, whereas the other gives more importance to delivery time. If these agents were to communicate their evaluations then, despite having a shared ontology, they would not be meaningful to the other agent. While a single interaction is generally not considered enough to base a trust evaluation on, such differences in how evaluations are computed are propagated all throughout the model, and eventually two syntactically equal evaluations can *mean* something different to different agents. Therefore, despite the work that has been done on applying common ontologies, for instance in the ART testbed [12], the scope in which this is possible seems limited.

2.1 Learning a Translation

The first work to address trust alignment directly is, to our knowledge, [13]. This work describes a trust model that evaluates a trustee with an integer between 1 and 4, where 1 stands for *very untrustworthy* and 4 for *very trustworthy*. The alignment process uses the recommendations from another agent about *known* trustees to calculate four separate biases: one for each possible trust value. First the alignment method calculates the own trust evaluations of the corresponding trustee for each incoming recommendation. The *semantic distance* between the own and other's trust is simply the numerical difference between the values of the trust evaluations. The semantic distances are then grouped by the value of the corresponding received trust value, resulting in four separate groups. Finally the bias for each group is calculated by taking the mode of the semantic distances in the group, resulting in four integers between -3 and 3, which can be used when the agent receives recommendations about unknown trustees. Simply subtract the corresponding bias from the incoming trust evaluation to translate the message. While this is a very simple approach it seems to work surprisingly

well. We will return to this method in Section 3.4, however first we will discuss later developments, based on a similar concept but recognizing that trust evaluations may differ between situations and thus the evidence for such trust evaluations must be taken into account in the translation.

2.2 Machine Learning Using Context

Current methods to learn a translation take the context into account, by using machine learning techniques to learn which own trust evaluation corresponds to a recommendation, when taking the evidence supporting the evaluations into account [14,15]. The evidence in these methods is a description in a shared, objective language of the interactions a trust evaluation is based on. For instance, the experimentation described in [14] links an evaluation of a sale interaction with a single propositional variable describing that sale (specifically whether the item was delivered on time or not). The alignment method uses this linked information: the evidence together with both its own and the other's trust evaluation as input for a machine learning algorithm. This algorithm learns a generalization, which serves to translate future communications in which the receiving agent cannot calculate its own trust evaluation, because the interaction being described is not shared. Insofar as we know there are two approaches which have been shown to work using this technique: BLADE [14] uses a conjunction of propositions to describe the interactions and a Bayesian Inference Learner to learn a generalization and we proposed a method [15] that allows a description of the interactions in first-order logic and an Inductive Logic Programming learner to find the generalization. While these two methods use different machine learning techniques, the largest difference between the two approaches is the representation of the contextual information. BLADE uses a propositional representation, which cannot adequately represent domains involving multiple entities and the relationships among them [16], while first-order logic is suited for this task. We discuss these methods in greater detail in Section 3.4.

3 Experiments

All the methods in the previous section have been implemented, however thus far no attempt has been made to show what approach is best used. As such it is an open question whether taking the context into account improves the alignment. Moreover, it has not been evaluated to what extent alignment methods improve communication at all. In this section we answer these questions empirically.

3.1 Experimental Setup

The aim of the experiments is to measure the effect of communication about trust on the accuracy of agents' trust evaluations. We are explicitly not interested in evaluating trust models and whether they choose the correct target. For this there are other methods, such as the aforementioned ART testbed. To measure

the effect of communication we need to compare two situations: (1) an agent's *estimated* trust evaluations, initially given incomplete information about the environment, but allowing communication about trust, and (2) that same agent's *most accurate* trust evaluations; given perfect and complete information about an environment. This allows for the comparison between the two evaluations and gives a measure for the *accuracy* of the estimated trust evaluation. By varying the amount of communication allowed and the type of alignment used we can measure the influence that alignment has upon the accuracy of the agents' trust evaluations.

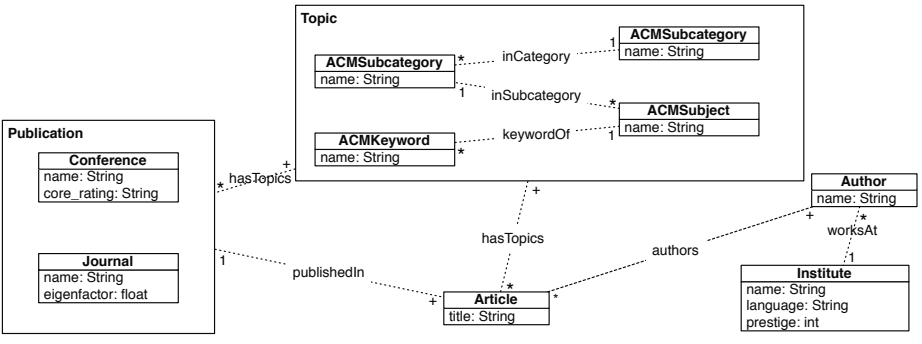

Fig. 1. A UML-like specification of the domain language

Drawing from the LiquidPub project [17], the experiments are focused around a scenario[1] in which agents have to recommend authors to each other, basing these recommendations on the articles they have written. We generate synthetic articles written by between one and five authors each. We specify these articles in a language using a fixed vocabulary, given in Figure 1, that describes properties of the articles. We consider these articles as representations of an interaction between authors, readers and any other stakeholders. We focus on the way readers observe such an interaction through the action of reading the article and forming an opinion about it and the authors. The authors are the trustees to be evaluated, the readers the evaluators and the articles serve as evidence.

In an initialization phase, the articles are divided over the reader agents, such that each reader only receives articles written by a configurable percentage of the author agents. The goal is to give each reader only partial information, so that each of them has incomplete information about only some of the authors in the system, thus creating the need for communication. For this communication two languages are needed. The first is that in which subjective trust evaluations can be communicated. This has a fixed syntax, but the semantics are the subjective evaluations of each agent: the meaning of trust is dependent on each agent's trust model. Because of the fixed syntax all agents will agree on what type of

[1] Code and documentation can be downloaded at http://www.megaupload.com/?d=SJL2NLH9 with password: coopis

Algorithm 1. Abstract Trust Model

Input: $t \in Authors$, the target author to be evaluated
Input: $Articles$, a set of articles, written by t
Input: $Communicated\_Evaluations$, a set of communicated evaluations from other evaluator agents in the system
Input: $default\_eval$, a default trust evaluation, used in case no articles have been observed and no communicated evaluations have been received

```
if Articles ≠ ∅ then
    article_ratings := ∅
    foreach Article a ∈ Articles do
        article_ratings := article_ratings ∪ evaluate(t, a)
    trust_eval := aggregate(article_ratings)
else if Communicated_Evaluations ≠ ∅ then
    certainty := 0
    foreach Evaluation e ∈ Communicated_Evaluations do
        if certainty(e) ≥ certainty then
            certainty := certainty(e)
            trust_eval := value(e)
else
    trust_eval := default_eval
```

Output: trust$(t, trust_eval)$

trust evaluations are allowed, but *why* a trustee is evaluated with any specific value is subjective and this is what needs aligning. To describe the articles we use the same language as the one used to generate them.

After the initialization the experiment runs for n rounds, in which each round represents the opportunity for the readers to communicate. In each round the agents may pick one other reader agent to communicate with. A communication act may be: a request to either align, or to get the other's trust evaluation of a single author. After n rounds of communication a measure of each agent's individual accuracy is calculated and, averaging these individual measures, the score of the entire run is determined. This score can then be compared to runs with a different value for n or using different methods of alignment.

3.2 Trust Models

In the experiments, we use five different reader agents, each with its own trust model. All these models use the same general structure, given in Algorithm 1. The models distinguish between direct trust and communicated trust. If the reader has observed any articles written by the author T, it uses direct trust. This depends on the **evaluate** and **aggregate** functions to calculate a trust evaluation. If no articles have been observed, then communicated trust is used: each communicated evaluation has an uncertainty associated with it, which is dependent on the alignment method used. The agent selects the single communication with the highest certainty to use. If there are also no communicated evaluations available, then a default trust evaluation is used. This is a very basic

trust model and most models in the literature use a more sophisticated method of aggregating information from different sources (e.g. direct trust, reputation, communicated evaluations), however this model is sufficient to show the problems that arise if the agents do not align and to evaluate the different alignment methods. Sophisticated aggregation methods have large advantages at the individual level, because they allow for richer models and more predictive methods, however if two agents use different aggregation methods, it is hard to distinguish whether the difference in trust evaluation is because the agents use a different aggregation method, or because they use different aspects of the interactions.

Work has been done on learning aggregated values [18], however this work is not yet applicable to the more complicated aggregation methods used in modern trust models. The trust alignment methods described in Section 2 avoid this issue by aligning the ratings of individual interactions. The agents can then use their *own* aggregation method, thereby obviating the need to solve the more complex problem of finding an alignment after aggregation. For the **aggregate** we take the average of the article ratings, although as we just explained, an agent could equally well use a probabilistic method such as BRS [19] or a more social network oriented approach, such as Yu & Singh's model [20].

The **evaluate** function is where each of the reader's trust models differs. Based on the description of articles in the domain ontology given in Figure 1, each agent has a different way of calculating some values for subjective properties of the article, such as readability or originality. Based on these, the agent calculates the rating of the author, using a list of "if-then-else" rules in Prolog, such as the following:

```
evaluation(Target, Article, 5) :- authors(Article, Authors), member(Target, Agents),
                significance(Article, Sig), Sig > 0.7, originality(Article, Ori),
                Ori > 0.7, readability(Article, Read), Read > 0.7, !.
```

This rule states that if the target agent is an author of the article and the observations of significance, originality and readability are all greater than 0.7 then the evaluation of the author, based on that article has value 5. All five of the readers' trust models are comprised of such rules, but they only coincide in the structure. The trust models differ in the actual content of the rules, such as the values attributed to different combinations of the subjective properties. Furthermore, the way in which the subjective properties, such as readability, are calculated, is different.

Additionally, one of the readers distinguishes between the first and other authors, using a different set of rules for either case. Another reader distinguishes between articles published in journals and those published in conferences. This leads to five different models, with different complexities for the alignment between them.

3.3 Strategy

In addition to the trust model, each agent must have a strategy to choose what to do in each round. While we cannot focus too much on this in the scope of this article, we realize that this choice may have a large influence on the outcome

of the experiment. We therefore implement two strategies for comparison. The first is a simple random strategy. Each agent chooses an author at random. It then chooses a reader agent at random to ask about that author. If it has not previously aligned with that reader, rather than asking for the agent's evaluation, it asks to align. If it has already aligned, it asks for the other agent's evaluation of the chosen author.

The second strategy is a non-random strategy in which each agent first chooses the author it has the least certain evaluation of. We use a very simple notion of certainty: an agent's certainty is equal to the percentage of the author's articles that the agent has observed. This notion may not be particularly accurate (for instance, if the author has written only very few articles), but it is only a heuristic for selecting which author to obtain more information about. It does not affect the trust evaluation. After choosing the target author, it picks the reader agent that has the most observations of that target and whose opinion has not yet been asked. After choosing the author and evaluator agent, this strategy behaves the same as the random strategy: if the agent has already aligned with the chosen evaluator it asks for a trust evaluation and otherwise it asks to align. While there are many optimizations possible, they are also further distractions from the main tenet of this research. We do not doubt that there are ways of improving the strategy of choosing when to align or with whom to communicate, however the main idea is that if we can show that the trust evaluations are more accurate with alignment than without, performance should only improve if the strategy is optimized.

3.4 Alignment Methods

Before discussing the experiments in detail we need to introduce the trust alignment methods we compare.

Average Bias. Our first alignment method is a very simple method, which does not take the context into account. When aligning, it calculates the mean difference between the other's recommendations and the own trust evaluations and use this as a single bias. We will call this method the alignment using an *average distance bias*.

Abdul-Rahman & Hailes' Method (AR&H). AR&H's model cannot be applied directly, because it requires discrete values to calculate the bias. Because in our models the aggregated trust evaluation is the average of an author's ratings as the trust evaluation, we do not have discrete values. However, we can apply AR&H's alignment method at the level of the ratings of individual articles, which are discrete: specifically, in our experiment they are natural numbers between -5 and 5. Furthermore, because we use a real value for the trust evaluation we can refine the method slightly by using the mean, rather than the mode for each bias. Other than that slight refinement, the method applied is the same as that already described in Section 2.1.

Koster et al.'s method. The third alignment method we test is the one we proposed in [15], using a first-order Inductive Logic Programming (ILP) algorithm. This is one of the two methods designed thus far, based on machine learning algorithms, the other being BLADE [14], which uses a propositional Bayesian Inference Learner. Comparing these two methods is not straightforward, because of the difference in representation. In [21], it is demonstrated empirically that propositional logic decision tree learners (which are propositional ILP algorithms) and Bayesian Inference Learners perform approximately equally, although ILP algorithms perform computationally better in large problems. Unfortunately BLADE is not equipped to deal with the more complex problem we consider here, in which a first-order – rather than a propositional – logic is used to describe articles. To learn relations in this language would require a different, first-order Bayesian network, which falls outside the scope of this work.

The implementation of our method, which we will refer to as Koster et al.'s method, follows the description in [22], which uses the first-order regression algorithm TILDE [23] to learn an alignment. Regression is a form of supervised learning, in which the goal is to predict the value of one or more continuous target variables [24] from a (finite) set of cases. A first-order regression algorithm does this by using, in addition to the numerical cases, an additional description in first-order logic. A case in our situation is a numerical rating of an article, together with a description of that article, communicated using the ontology in Figure 1. The algorithm is implemented in the ACE package [25] and gives as output a set of Prolog clauses which can be used to translate future communications. The technical report describing this version of the alignment method includes some preliminary experimentation. It gives experiments showing under what circumstances the learning algorithm gives good results, but does not place this in a frame of reference in which the algorithm can be compared to other methods, or even with the lack of alignment.

3.5 Comparing Alignment Methods

The first experiment aims to compare the alignment methods with each other as well as with the two default modes: no communication at all and communication without alignment. As described in Section 3.2, if an agent has no knowledge of an author, it uses a default trust evaluation. Because the agents have incomplete information about the environment, this case will occur when no, or too little, communication is allowed. The default evaluation can be seen as the agent's initial evaluation of any author, before learning anything about it and we distinguish between the following options:

A mistrusting agent. always gives its most negative evaluation to any agent it has no knowledge of.

A trusting agent. always gives its most positive evaluation to any agent it has no knowledge of.

A neutral agent. always gives a middle evaluation to any agent it has no knowledge of.

A reflective agent. calculates the mean of all its previous trust evaluations of other agents and uses this for any agent it has no knowledge of.

The first three options give a fixed value, independent of the agent's evaluations of other targets in the system, whereas the last option allows the agent some type of adaptability, depending on what trust evaluations it has so far given to other targets. If the targets it has knowledge of are all bad agents, then it will be more similar to the first option, whereas if they are all good it will be more similar to the second. Of all options for no communication we expect this will be the best choice for an agent, although it is also the only option which requires extra computation.

Setting up the Experiment. We start by running a number of experiments to ascertain which parameters should be used for a fair comparison between the alignment models. By changing the total number of articles and the percentage of articles observed by each agent we can change the average number of articles shared by the agents. This mainly influences the functioning of AR&H's and Koster et al.'s methods. At low numbers of shared articles AR&H's method outperforms Koster et al.'s, however with around 100 articles shared between any two agents Koster et al.'s method starts to outperform AR&H's. This difference in performance increases until approximately 500 articles are shared, on average. Running the experiment at higher numbers of shared interactions is unnecessary, because all algorithms have reached peak performance. We opt to run our experiments with 500 shared articles, thus achieving the optimal results obtainable with each of the alignment methods. The goal of the experiment is to measure the influence the different alignment methods have. Therefore we require each agent's information about the environment to be incomplete. We achieve this by only allowing each reader agent to observe articles by 40% of the author agents. This means that to find out about the other 60% of the authors, communication is required. By having a total of 2000 articles written by different combinations of 50 authors, we can measure the influence of communication while still allowing agents to, on average, share 500 articles. We run each experiment 50 times with different articles to have a decent statistical sample. In this first experiment we vary two parameters: the number of rounds in which agents may communicate and the baseline trust evaluation an agent uses to evaluate targets it has no information of. The results are plotted in Figure 2. The y-axis represents the error with respect to the most accurate evaluation: if the agent were to have perfect information about all articles. Given the probability distribution of a trust model's evaluations, the error is the probability of the agent's evaluation of a trustee being between the estimated and most accurate evaluation[2]. It is a measure of the inaccuracy of the alignment method, because the percentage on the y-axis is not the chance that an agent's evaluation is wrong, but rather a measure of *how* wrong an agent is on average.

[2] Calculated as the cumulative probability between the two values.

Results. We firstly see in Figure 2(b) that if we use the neutral baseline (using 0 as the default evaluation), then all communication is preferable over no communication. The same is not true if we use the reflective baseline (taking the average of past evaluations of other targets), as seen in Figure 2(a). In this case communication without alignment gives worse results than not communicating at all. This is easily explained: if the observed articles are a representative sample of the population then the mean of trust evaluations based on these will be near the mean of the most accurate trust evaluations. Consequently, always using the default will be quite good. However, the other evaluators' trust evaluations are based on different properties of the articles and may thus be further from the most accurate trust evaluation. The more of these unaligned communicated evaluations an agent incorporates, the less accurate its evaluations will become. We allocate articles at random and therefore each agent does observe a representative sample of them. This same would not be true if the network were not a random network or the location of an agent in the network influenced its trustworthiness: the trustees observed would not be a representative sample of the other agents in the network and the error from using the default would be larger. If this error becomes large enough it would resemble the situation with the neutral baseline, in which case the error from using unaligned communications results in an improvement. We have omitted the experiments using the trusting and distrusting baselines, because their results are very similar to those of the experiment with the neutral baseline and thus add very little information.

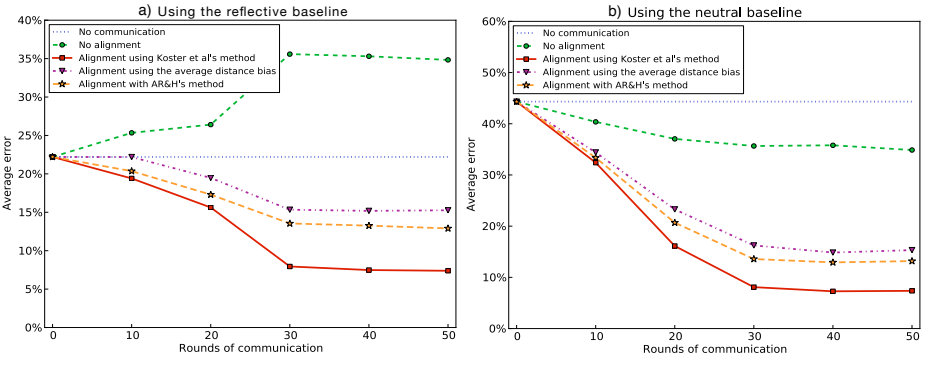

Fig. 2. Average score - with and without alignment

The main result of this experiment is that communication *with* alignment *always* gives significantly better results than either no communication or communication without alignment. In the graphs of Figure 2 we have plotted the average accuracy for all five of the agents, however as discussed in Section 3.2, the individual trust models play a large role in this performance. The different alignment methods give different returns for the individual agents, but always significantly outperform the situations without alignment. Furthermore the differences seen in the graphs are significant. Because the accuracy measure is

not normally distributed we evaluated this by using a Kruskal-Wallis test for analysis of variance [26]. The pair-wise difference is also significant, as tested using Mann-Whitney U-tests[3]. While this seems to indicate that Koster et al.'s method performs slightly better than either of the methods which do not take the context into account, it seems premature to draw this conclusion, given the assumptions underlying the experiment. However, this experiment does serve to show that *some* form of alignment is necessary to communicate about trust. The real advantages of taking the context into account are discussed in Section 3.7, where we deal with untrustworthy communicators.

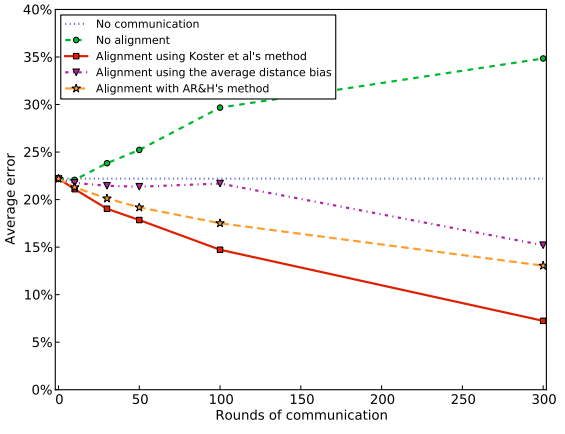

Fig. 3. The random strategy for partner selection

3.6 Using a Random Strategy

The first variation on this experiment we explore is to change the strategy for selecting a communication action. The first experiment uses the non-random strategy and we compare these results to the exact same experiment, but using the random strategy. For this experiment we use the reflective baseline and the results up to 300 rounds of communication are plotted in Figure 3. As is to be expected, we see that in the short term picking the communication at random does quite significantly worse than using a heuristic to choose whom to communicate with: after 50 rounds of using the non-random strategy all alignment methods are doing significantly better (see Figure 2(a)) than after 50 rounds of using the random strategy (Figure 3). However in the long run the effect is flattened out and eventually the random strategy achieves the same optimum alignment as the non-random strategy. This implies that, after enough rounds of communication, the optimum is fixed by the alignment method and the strategy does not influence it. To show that the value they converge on really is the lowest average error an agent can achieve using the given alignment method, we run

[3] For all tests we obtain $p \ll 0.01$: the probability that the different datasets were obtained from the same population is very small.

the non-random strategy for 150 rounds, which is enough rounds for all possible communications to take place. For all the methods tested we compare this with the outcome after 50 rounds for the non-random strategy and 300 rounds for the random strategy: these values are mutually indistinguishable[4], showing that even after exhausting all possible communication the alignment is not further improved and truly is an optimum.

The strategy, however, does have a strong influence on how fast this optimum is reached. Using a different strategy will change the speed of convergence, but any good strategy will allow agents to converge on the most accurate evaluations of all agents in the system, just better strategies will converge faster.

This means that from an agent designer's viewpoint the strategy and alignment method can be completely separated: if an evaluator agent requires information about a target agent, the alignment method defines an optimal accuracy for this information while the strategy defines how many agents on average the evaluator agent must communicate with before it has communicated with the agent giving the most accurate information.

3.7 Simulating Lying Agents

In the first experiment we tacitly assumed all agents are truthful and willing to cooperate. If they do not cooperate with the alignment process there is obviously nothing we can do, but assuming other agents are truthful is a rather strong assumption. This experiment is therefore set up to see what happens with the communication if we simulate the other agents passing incorrect information. Note that if the agents are entirely consistent in their lies, AR&H and Koster et al.'s alignment methods will be able to deal with this perfectly, as they learn a translation from the other's trust evaluation. Additionally, Koster et al.'s method is even able to deal with lying if it is not always consistent, but based on some specifics of the underlying article (such as: always lie if the author works at a certain institute). The problem for all alignment algorithms appears if agents just invent a random value. We run another round of experiments, this time increasingly replacing truthful agents by lying ones. A lying agent, rather than giving an actual trust evaluation, communicates random ratings of articles. The results can be seen in Figure 4. The agents using communication use the reflective baseline as their default evaluation in the case they do not have other information available.

Results. We focus first on graph (d) in Figure 4 and see that if all agents are lying then communication with no alignment converges to the accuracy of the trust evaluations without communications and using the average of all possible trust evaluations as the fixed evaluation for unknown agents. We can explain this convergence by seeing that the mean of all possible trust evaluations is also the mean value of a random distribution over the possible trust values. A similar thing happens using AR&H's method, which calculates what its own trust evaluation should be if the other agent communicates a certain value. However,

[4] Obtaining $p \gg 0.05$ for all Mann-Whitney U-Tests.

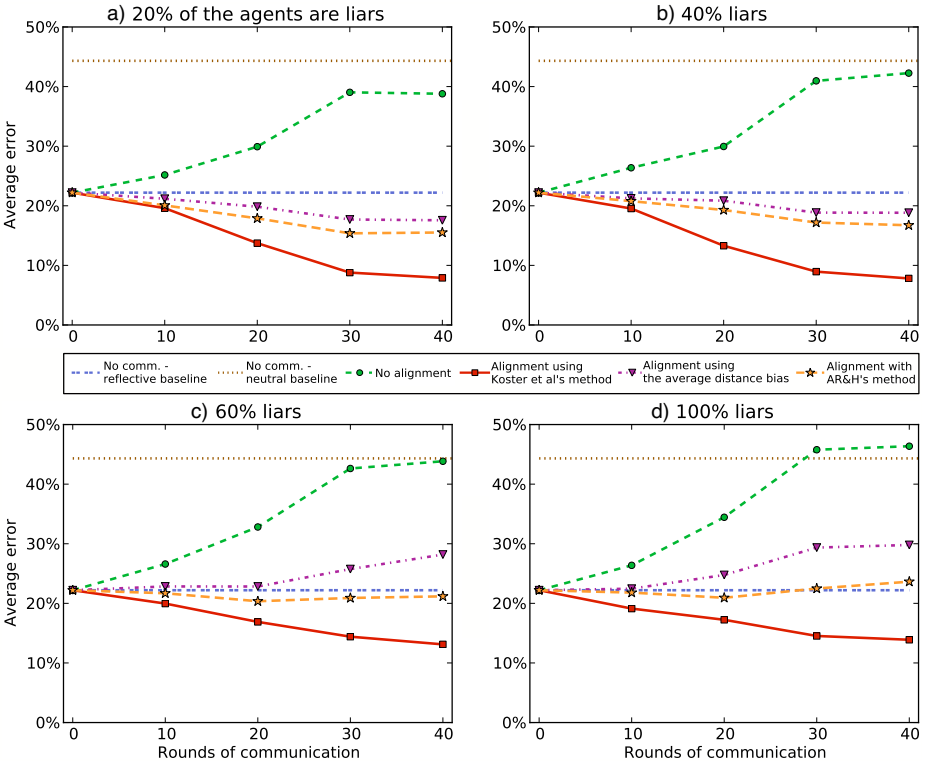

Fig. 4. Slow degradation from a domain with no lying agents to a domain with all lying agents

because the other's trust evaluations are random, choosing all those at a certain value will give a random sample of the own trust evaluations, the mean of which will, on average, be the mean of all the own trust evaluations, so AR&H's model stays approximately flat on the default baseline (using the average of all the agent's own trust evaluations). For similar reasons the average bias does slightly worse, converging to a value between the two baselines. Koster et al.'s method, on the other hand, appears to hardly be affected by the noisy trust evaluations. This shows a large advantage of taking the context into account: Koster et al.'s method maintains its performance, because the communications in the domain language can be used for the alignment method to compensate for the noisy trust evaluations. It ignores the noisy trust evaluations and learns by using *only* the information about the underlying articles. If we were to add noise to this part of the communication as well, Koster et al.'s model would collapse to AR&H's and thus stay flat as well.

With this explanation of what happens when all agents lie we can see that by slowly adding more liars to the system, the performance of the various algorithms morphs from the system with no liars (Figure 2(a)) to the system with all liars (Figure 4(a)-(d) progressively). To prevent this from happening a further

refinement would be necessary: detecting which agents are the liars and disregarding their communications, as discussed in Section 2.

4 Discussion

The experimentation in the previous section demonstrates that trust alignment improves the accuracy of agents' trust evaluations. Koster et al.'s method even works in situations where the communicated evaluations are 100% noise. However, we must take care when interpreting these experiments. The first thing to note is that the trust models used, as described in Section 3.2, are simplifications of those used in the literature. Agents only communicate the evaluations based on their own direct experiences, rather than having an evaluation which is aggregated from a number of different sources. This, however, only strengthens the point we are trying to make: the more complex an agent's trust evaluation can be, the greater the probability that two agents, despite using the same ontology for their trust evaluations, *mean* different things, because the actual way they calculate the evaluations are completely different. The use of more complex trust models thus leads to an even greater need for alignment. Unfortunately, the more complex the trust models, the more information will be required to apply a method such as Koster et al.'s, which requires numerous samples of different types of evidence supporting the trust evaluations. Luckily, the worst case for Koster et al is that the domain information is too complex to use, in which case it will perform similarly to AR&H's method. In such cases there may be other machine learning techniques, such as case based reasoning [27], which is designed to handle large sets of complex data, which could offer a solution.

Additionally the alignment is required to take place before aggregation. This means that regardless of how complex the aggregation method is, as long as what is being aggregated is not too complex, the alignment can work. However, it also means that a large amount of information needs to be communicated. There may be scenarios in which this communication is prohibitive and a simpler form of alignment, such as AR&H's method, or even the average bias, must be used. However, in domains such as e-Commerce, a lot of data is readily made available: on eBay[5] for example, for any transaction it is public knowledge what item was sold and how much it cost. Similarly in social recommender systems, which is how we would classify the example scenario in this paper, people are often willing to explain their evaluation of an experience in great detail (such as on Tripadvisor[6]). This is exactly the type of information that is needed for aligning. If necessary this could be combined with a method of incentivizing truthful feedback, such as described in [28]. This could also be helped to mitigate lies, which is the final point for discussion.

Our model only generates noise in the trust evaluation, not in the description of the evidence. Furthermore, if a hostile agent has knowledge of the aligning agent's trust model, it could tailor its alignment messages so that it can send

[5] www.ebay.com
[6] www.tripadvisor.com

false evaluations undetected. Luckily a lot of work has been done in detecting fraudulent, or inconsistent information, both in the context of trust and reputation [7,8], as well as in collaborative filtering [29]. As briefly mentioned in Section 3.7 such a method could be used in combination with alignment methods. By merging an alignment method with a filtering method the efficacy of both can be significantly improved. Good alignment rules can be used to minimize the useful information discarded, while the filtering methods are well equipped to decide when an agent is not giving any useful information at all.

5 Conclusions and Future Work

The experimentation shows clearly that communication without alignment may have a negative influence on the accuracy of an agent's trust evaluations and thus that alignment is a necessary step when talking about trust. We see that even a simple alignment method such as calculating an average bias, can give a significant boost to the trust model's accuracy. AR&H and Koster et al.'s methods function at least as well as not communicating even if all other agents are liars. Koster et al.'s method outperforms all other methods tested, by taking *the context* in which a trust evaluation was made into account. This performance, however, comes at a cost: Koster et al.'s model uses a relatively complex learning algorithm and requires communication about not only ratings of individual interactions, but also an objective description of the interaction it is based on. The functioning of this alignment method may very well depend on the expressiveness of the language for describing interactions. If such a language is very basic, then alignment may not be possible and a simpler method must be used. Similarly, privacy issues may arise in scenarios where agents are willing to exchange trust evaluations, but not anything more. In such cases the best we can do is the method taking an average bias. Whether the increased complexity and communication load is worth the added performance should be evaluated per domain. Additionally, the trust models themselves influence the accuracy of alignments. Analyzing the interplay of some different trust models used in practice, as well as more – or less – descriptive domain languages for describing the context, is an important concern for future research. Another promising method for alignment lies in argumentation about trust [30]. Such methods attempt to establish and explain the causal link between what happened in the environment and the trust evaluation it resulted in by giving a formal argumentation framework in which agents can communicate their reasons for trust. Thus far such methods are not yet developed sufficiently to be applied for alignment.

Acknowledgements. This work is supported by the Generalitat de Catalunya grant 2009-SGR-1434 and the Spanish Ministry of Education's Agreement Technologies project (CONSOLIDER CSD2007-0022, INGENIO 2010) and the CBIT project (TIN2010-16306). Additionally the authors would like to thank the anonymous reviewers for their feedback.

References

1. Jøsang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. Decision Support Systems 43(2), 618–644 (2007)
2. Conte, R., Paolucci, M.: Reputation in Artificial Societies: Social beliefs for social order. Kluwer Academic Publishers (2002)
3. Castelfranchi, C., Falcone, R.: Trust Theory: A Socio-cognitive and Computational Model. Wiley (2010)
4. Koster, A.: Why does trust need aligning? In: Proc. of 13th Workshop "Trust in Agent Societies", Toronto, pp. 125–136. IFAAMAS (2010)
5. Euzenat, J., Shvaiko, P.: Ontology matching. Springer, Heidelberg (2007)
6. Schorlemmer, M., Kalfoglou, Y., Atencia, M.: A formal foundation for ontology-alignment interaction models. International Journal on Semantic Web and Information Systems 3(2), 50–68 (2007)
7. Teacy, W.T.L., Patel, J., Jennings, N.R., Luck, M.: Travos: Trust and reputation in the context of inaccurate information sources. Journal of Autonomous Agents and Multi-Agent Systems 12(2), 183–198 (2006)
8. Şensoy, M., Zhang, J., Yolum, P., Cohen, R.: Context-aware service selection under deception. Computational Intelligence 25(4), 335–366 (2009)
9. Pinyol, I., Sabater-Mir, J.: Arguing About Reputation: The lRep Language. In: Artikis, A., O'Hare, G.M.P., Stathis, K., Vouros, G.A. (eds.) ESAW 2007. LNCS (LNAI), vol. 4995, pp. 284–299. Springer, Heidelberg (2008)
10. Casare, S., Sichman, J.: Towards a functional ontology of reputation. In: AAMAS 2005: Proc. of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, Utrecht, The Netherlands, pp. 505–511. ACM (2005)
11. 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 2009, Budapest, Hungary (2009)
12. Brandão, A.A.F., Vercouter, L., Casare, S., Sichman, J.: Exchanging reputation values among heterogeneous agent reputation models: An experience on art testbed. In: Proc. of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007), Honolulu, Hawaii, pp. 1047–1049. IFAAMAS (2007)
13. Abdul-Rahman, A., Hailes, S.: Supporting trust in virtual communities. In: Proceedings of the 33rd Hawaii International Conference on System Sciences, vol. 6, pp. 4–7 (2000)
14. Regan, K., Poupart, P., Cohen, R.: Bayesian reputation modeling in e-marketplaces sensitive to subjectivity, deception and change. In: Proceedings of the 21st National Conference on Artificial Intelligence (AAAI), Boston, MA, USA, pp. 1206–1212. AAAI Press (2006)
15. Koster, A., Sabater-Mir, J., Schorlemmer, M.: Engineering trust alignment: a first approach. In: Proc. of the Thirteenth Workshop "Trust in Agent Societies" at AAMAS 2010, Toronto, Canada, pp. 111–122. IFAAMAS (2010)
16. De Raedt, L.: Logical and Relational Learning. Springer, Heidelberg (2008)
17. Liquid publications: Scientific publications meet the web. September 2, (2010), `http://liquidpub.org`
18. Uwents, W., Blockeel, H.: A Comparison Between Neural Network Methods for Learning Aggregate Functions. In: Boulicaut, J.-F., Berthold, M.R., Horváth, T. (eds.) DS 2008. LNCS (LNAI), vol. 5255, pp. 88–99. Springer, Heidelberg (2008)

19. 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)
20. Yu, B., Singh, M.P.: An evidential model of distributed reputation management. In: AAMAS 2002: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 294–301. ACM, New York (2002)
21. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine Learning 29(2–3), 131–163 (1997)
22. Koster, A., Sabater-Mir, J., Schorlemmer, M.: Engineering trust alignment: Theory and practice. Technical Report TR-2010-02, CSIC-IIIA (2010)
23. Blockeel, H., De Raedt, L., Ramon, J.: Top-down induction of clustering trees. In: Shavlik, J. (ed.) Proceedings of the 15th International Conference on Machine Learning, pp. 55–63 (1998)
24. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2006)
25. 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, 135–166 (2002)
26. Corder, G.W., Foreman, D.I.: Nonparametric Statistics for Non-Statisticions: A Step-by-Step Approach. Wiley (2009)
27. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Communications 7(1), 39–59 (1994)
28. Witkowski, J.: Truthful feedback for sanctioning reputation mechanisms. In: Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI 2010), Corvallis, Oregon, pp. 658–665. AUAI Press (2010)
29. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Advances in Artificial Intelligence 2009. Article no. 421425 (January 2009)
30. Pinyol, I., Sabater-Mir, J.: An argumentation-based protocol for social evaluations exchange. In: Proceedings of The 19th European Conference on Artificial Intelligence (ECAI 2010), Lisbon, Portugal (2010)