A Statistical Relational Model for Trust Learning

Achim Rettinger
Department of Informatics
Technical University of Munich
85748 Garching, Germany
achim.rettinger@cs.tum.edu

Matthias Nickles
Department of Computer Science
University of Bath
Bath BA2 7AY, UK
m.l.nickles@cs.bath.ac.uk

Volker Tresp
Corporate Technology
Siemens AG
81739 Munich, Germany
volker.tresp@siemens.com

ABSTRACT

We address the learning of trust based on past observations and context information. We argue that from the truster's point of view trust is best expressed as one of several relations that exist between the agent to be trusted (trustee) and the state of the environment. Besides attributes expressing trustworthiness, additional relations might describe commitments made by the trustee with regard to the current situation, like: a seller offers a certain price for a specific product. We show how to implement and learn context-sensitive trust using statistical relational learning in form of the Infinite Hidden Relational Trust Model (IHRTM). The practicability and effectiveness of our approach is evaluated empirically on user-ratings gathered from eBay. Our results suggest that (i) the inherent clustering achieved in the algorithm allows the truster to characterize the structure of a trust-situation and provides meaningful trust assessments; (ii) utilizing the collaborative filtering effect associated with relational data does improve trust assessment performance; (iii) by learning faster and transferring knowledge more effectively we improve cold start performance and can cope better with dynamic behavior in open multiagent systems. The later is demonstrated with interactions recorded from a strategic two-player negotiation scenario.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems, Intelligent agents*
; I.2.6 [**Artificial Intelligence**]: Learning—*Analogies*

General Terms

Algorithms, Design, Experimentation, Security, Theory

Keywords

Computational Trust, Trust Modeling, Relational Learning

1. INTRODUCTION

The need for predicting an agent's future behavior is getting increasingly important in distributed systems since contemporary developments such as the Semantic Web, Service Oriented Architecture, as well as Pervasive, Ubiquitous and Grid Computing are applied mainly to open and dynamic systems with interacting autonomous agents. In many situations, such agents show a highly contingent behavior, and often it is not feasible to implement effective mechanisms to enforce socially fair behavior as pursued in mechanism design or preference aggregation. A potential solution to these problems is the transfer of the human notion of trust to a machine-computable model, realizing computational trust.

However, in our opinion human-trust and its main objective as a mechanism for complexity reduction in uncertain and unknown situations still has not yet found an equivalence in computational trust. The essential property of such trust-situations is that the human respectively the agent does *not* have sufficient information that can directly be applied to assess a trust value. Instead trust is inferred from related contextual factors. We think that this lack of a certain basis for decision-making is *the* defining characteristic of trust and is not taken into account in most current trust models. Consequently, situations where trust can *exclusively* be based on the reputation calculated from recommendations or trust-networks does not comply with this strict specification of trust. The same holds for cognitive and game theoretic models of trust based on computable incentives of the trustee or statistical models dependent on repetitive interactions in a restricted context-*independent* environment.

Based on those observations our objective is to relax existing restrictions of computational trust by trying to learn trust in a rich context-dependent relational environment: Modeling the environment *from the perspective of the truster*, two entities, both described by their respective attributes, constitute a trust situation: (i) the trustee and (ii) the state of the environment. Most importantly, both entities are interconnected by relational dependencies.

If the trustworthiness depends not only on the trustee but also on the state of the environment in which one needs to trust, the truster can make more precise decisions and can apply learned knowledge to a wider range of situation. For instance, a seller might be trustworthy if offering a specific product, but not when offering another product. Furthermore, in such a situation a relation like the price might help to assess trustworthiness while depending on a particular product and the seller at the same time. By taking all this into account, we can improve predictions, give more meaning to trust and at the same time - by generalizing from different contexts - increase learning efficiency.

In the following we show how to implement and learn

Cite as: A Statistical Relational Model for Trust Learning, Achim Rettinger, Matthias Nickles and Volker Tresp, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16., 2008, Estoril,Portugal,pp. 763-770.

context-sensitive relational trust using one specific statistical relational model. Our Infinite Hidden Relational Trust Model (IHRTM) is based on recently introduced *infinite relational models* (see [12] and [4]). The practicability and effectiveness of this approach is evaluated empirically on user-ratings gathered from eBay. Our results suggest that (i) the inherent clustering achieved in the algorithm allows the truster to characterize the structure of a trust-situation and provides meaningful trust assessments (see Section 4.2); (ii) utilizing the collaborative filtering effect associated with relational data does improve trust assessment performance (see Section 4.3); (iii) by learning faster and transferring knowledge more effectively we improve cold start performance and can cope better with dynamic behavior in open multiagent systems. The later is demonstrated with interactions recorded from a strategic two-player negotiation scenario (see Section 4.4).

The next section introduces the statistical relational representation used for context-dependent trust modeling accompanied by an intuitive illustration of modeling transactions and feedback on eBay. Section 3 describes the technical details and the inference algorithm used to calculate cluster assignments and trust values in the special case of using our IHRTM. Section 4 presents the experimental analysis on three different levels: First the clustering effect on the eBay data is evaluated, then the predictive performance is compared to propositional learning algorithms, and finally the learning efficiency is demonstrated on data from an automated negotiations scenario. Section 5 discusses related work, and Section 6 outlines future research directions and provides conclusions.

2. MODEL DESCRIPTION

Relational models are an obvious formalization of requirements arising from the relational dependencies of entities in social, biological, physical and many other systems.

Our Infinite Hidden Relational Trust Model (IHRTM) consists of two entity classes: On the one hand the trustee-agent a and on the other hand specific elements of the state s of the environment. Both entities can be equipped with attributes Att^a and Att^s, respectively. The interdependencies are expressed as relation $interacts(a, s)$ with attributes Att^c (commitment) and Att^t (trust). Figure 1 illustrates the IHRTM as a DAPER model (c.f. [3]). Entity classes are depicted as rectangles and the relationship class as a rhombus. Observable evidence Att is modeled in attribute classes of entities and relationships (ovals). As in a classical non-relational Bayesian network, direct statistical dependencies are modeled as directed arcs. The DAPER model should be thought of as a template which, based on the actual objects in the domain, is expanded into the ground Bayesian network.

To illustrate the abstract model we will use the *eBay feedback-system* as a concrete example throughout this paper. Being the most popular online auction and shopping website, fraud on eBay is a serious and well-known problem. An attempt to deal with fraud is the eBay feedback-system where users leave feedback about their past transactions with other eBay-users.

Suppose the truster-agent is a buyer who wants to build a context-sensitive relational trust model to analyze the trust situation on eBay in general and assess trust values for purchases from eBay in particular. In this scenario, the truster

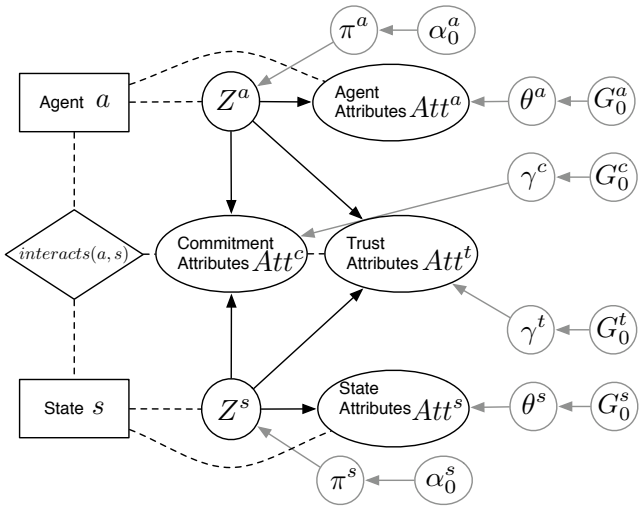

Figure 1: Graphical Representation 1: DAPER Model

itself does not need to be modeled explicitly because he learns a *personalized* model based on its own viewpoint and experience. The trustee a however represents sellers on eBay and the state s represents items that are for sale. The relation $interacts(a, s)$ would best be specified as $offers(a, s)$ in this context.

The attributes Att specify the observable features of the trust situation. Att^a describes properties of the seller like the feedback score, the percentage of positive feedback and his length of membership. Att^s specifies features that are associated with the product, for instance its category and its condition (new or used). The price however is represented as a relational attribute Att^c because a different seller could offer the same product for a different price. Thus, Att^c stands for all *commitments* seller and buyer make in the negotiation process. Besides the price or winning bid this can e.g. be shipping costs, bidding history, extent of warranty, payment details and shipping rates. Finally Att^t can include all dimensions of trust that are important for the truster when he finally gives feedback about his purchases. Relevant dimensions might be: actual shipping time, whether the item was as described, if the communication with the seller was as expected and so on.

As an example, one could now express the trustworthiness of an offer concerning product quality Att^t, given the seller a offers item s for price Att^c. Note that more than one attribute per entity or relation can be considered as well.

3. TECHNICAL DETAILS

To complete the technical details of our specific relational trust model we now introduce the remaining elements of the IHRTM. Following the ideas of [12] and [4] we assign to each entity a hidden variable, denoted as Z^a and Z^s and depicted as circles in figure 1. Related to the hidden states in hidden Markov models, they can be thought of as unknown attributes of the entities and are the parents of both the entity attributes and the relationship attributes. The underlying assumption is that if the hidden variables were known, both entity attributes and relationship attributes can be well predicted. A very important result of introducing the hid-

den variables is that now information can propagate in the ground network, which consists here of attribute variables exchanging information via a network of hidden variables.

Given that the hidden variables Z have discrete probability distributions they intuitively can be interpreted as cluster variables where similar entities (similar sellers or similar items) are grouped together. The cluster assignments (or hidden states) of the entities are decided not only by their attributes, but also by their relations. If both the associated seller and item have strong known attributes Att^a and Att^s, those will determine the cluster assignments and the prediction for Att^t. In terms of a recommender-system terminology we would obtain a content-based recommendation system. Conversely, if the known attributes Att^a are weak, then the cluster assignments Z^a for the seller a might be determined by the relations to items s and cluster assignments of those items cluster assignments Z^s. Accordingly, this applies to items s and its cluster assignment Z^s. In terms of a recommender-system terminology we would obtain a collaborative-filtering system. Consequently, IHRTM provides an elegant way to combine content-based predictions with collaborative-filtering prediction.

In the IHRTM, Z has an infinite number of states. Mixture models with infinite number of states are Dirichlet process (DP) mixture models, which have the property that the number of actually occupied components is determined automatically in the inference process. The fundamental idea is, that depending on the complexity of the problem, the model can "decide" for itself about the optimal number of states for the hidden variables; thus a time consuming optimization of the number of clusters can be avoided.

After sketching the functioning of the infinite hidden variables, we can complete the model by describing the local distribution classes denoting the parameters and hyperparameters of the probability distributions. They are shown as small gray circles in the DAPER model (figure 1). As an alternative to the DAPER model, we can display the structure of the IHRTM as a plate model, another commonly used graphical representation for statistical relational models (figure 2).

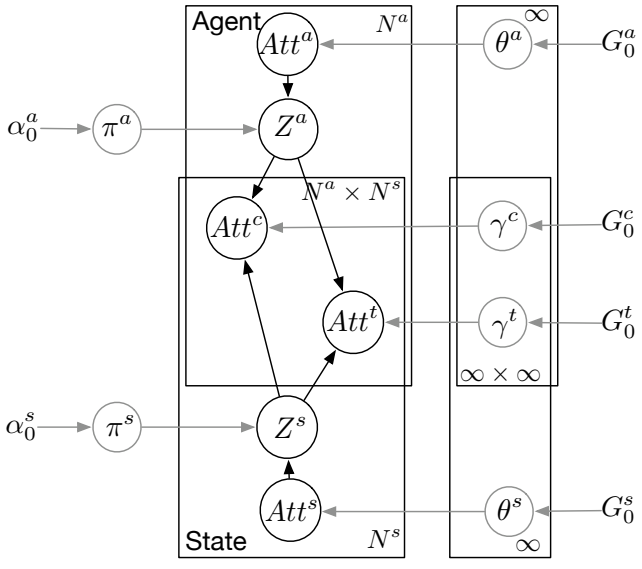

Figure 2: Graphical Representation 2: Plate Model

Now we consider the variables for the seller entity. For each specific seller i there is a hidden variable Z_i^a with the flexible and potentially infinite number of states K^a. The clustering $Z_i^a = k$ specifies the assignment of seller i to the specific cluster k. The weights $\pi^a = (\pi_1^a, ..., \pi_{K^a}^a)$ are multinomial parameters with $P(Z^a = k) = \pi^a$ and are drawn from a conjugated Dirichlet prior, $\pi^a \propto Dir(\cdot|\alpha_0^a, \alpha^a)$. $\alpha^a = (\alpha_1^a, ..., \alpha_{K^a}^a)$. α_k^a represents our prior expectation about the probability of a seller being in cluster k. $\alpha_0^a > 0$ determines the tendency of the model to either use a large number (large α_0^a) or a small number of clusters in Z (small α_0^a). For additional technical details on Dirichlet process mixture models, consult for example [10].

Since we only consider discrete attributes in our eBay example, a particular attribute Att^a is a sample from a multinomial distribution with multinomial parameters $\theta^a = (\theta_1^a, ..., \theta_{K^a}^a)$. The base distributions G_0^a and G_0^s are the associated conjugate priors. So, $\theta^a \propto G_0^a$. The same applies to the multinomial parameter γ for each of the $K^a \times K^s$ configurations related to each relational attribute Att^c and Att^t. Again, a Dirichlet process prior is employed, so that $\gamma^c \propto G_0^c$.

Now we briefly describe the generative models for the IHRTM. The method we use to generate samples from a Dirichlet Process mixture model is the Chinese restaurant process (CRP, see [10]). The clustering of data points in a DP can be explained by the following analogy: Imagine a restaurant with an infinite number of tables. Now customers enter the restaurant one by one and choose a table to sit down. Each customer either chooses to sit down at an unoccupied table or join some other customers at an already occupied table, where the table selection probability is proportional to the number of persons already sitting at a table. Applying this scenario to the Dirichlet process, the tables are clusters and the customers are data-points. After N data-points are sampled the $N + 1^{th}$ sample is generated as follows.

- The $N + 1^{th}$ agent is assigned to an existing agent cluster i with probability $\frac{N_i}{N+\alpha_0}$ and inherits parameters θ_i and γ.
- With probability $\frac{\alpha_0}{N+\alpha_0}$ the agent is assigned to a new cluster $K + 1$.
  For the new user cluster, new parameters θ_i and γ are generated as described above.

The procedure is repeatedly applied to all hidden variables in the ground network.

3.1 Inference

Based on the generative model presented in the previous section we can now generate samples from the IHRTM. In particular, we are interested in how to generate samples from the unknown states and parameters, given observed data. The most important goal is to infer the conditional distribution of the hidden variables Z^a, Z^s given all known attributes entity attributes Att^a and Att^s as well as relationship attributes Att^c and Att^t. This eventually allows us to make predictions about unknown attributes, like target value Att^t.

A way to approximate this posterior distribution of the hidden variables is by means of Gibbs sampling (GS), an MCMC-method. In our model, it is possible to formulate a

GS in which only samples from the hidden variables are generated by integrating out model parameters (see [12]). The Markov chain is thus defined only for the hidden variables of all entities in the given domain. The GS iteratively samples the hidden variable Z^a, conditioned on the other hidden variables Z^s until the procedure converges. In particular, Z is updated as:

1. For Z^a: Pick a random agent i . Assume that for N_k^a agents, $Z^a = k$ without counting user i.

   Either assign agent i to cluster k with probability proportional to

   $$P(Z_i^a = k | Z_{j\neq i}^a, Att_i^a, \theta^a, \gamma^c, \gamma^t, Z^s) \propto kP(Att_i^a | \theta_k^a, \gamma_{k,*}^c, \gamma_{k,*}^t)$$

   where N_k is the number of agents already assigned to cluster k and $\gamma_{k,*}$ notes the relation parameters of agent cluster k and all state clusters.

   Or generate a new cluster $K+1$ with probability proportional to

   $$P(Z_i^a = K^a + 1 | Z_{j\neq i}^a, Att_i^a, \theta^a, \gamma^c, \gamma^t, Z^s) \propto \alpha_0^a P(Att_i^a | \theta_k^a, \gamma_{k,*}^c, \gamma_{k,*}^t)$$

2. For Z^a: Pick a random state j and update its cluster assignment Z^s, accordingly.

3. If during sampling a state becomes unoccupied, remove that state from the model and re-assigned indices.

After the Markov chain has converged, standard statistical parameter estimation techniques can be used for estimating the parameters γ_{k^a,k^s}^t of Att^t from given cluster assignments. For a detailed description of the algorithm we refer to [12]. We extended the algorithm, as just described, to enable the handling of more than one relationship attribute. Being able to use an arbitrary number of relationships is essential to enable a rich representation of the interaction context as well as multidimensional trust values.

3.2 Implications

The ultimate goal of the model is to group entities into clusters. A good set of partitions allows to predict the parameters γ of attributes Att^c and Att^t by their mere cluster assignments. In the ground truth, our model assumes that each entity belongs to exactly one cluster. It simultaneously discovers clusters and the relationships in-between clusters that are best supported by the data, ignoring irrelevant attributes.

Although the value of attributes is determined entirely by the cluster assignment of associated entities, there is no need for direct dependencies between attributes or extensive structural learning. The cluster assessment of a entity is influenced by all corresponding attributes and cluster assessments of related entities. This way information can propagate through the whole network while the infinite hidden variables Z act as "hubs". This allows for a collaborative filtering effect. Cross-attribute and cross-entity dependencies can be learned, something which is not possible with a "flat" propositional approach that assumes independent and identical distributed (i.i.d.) data.

At the same time the number of clusters does *not* need to be fixed in advance. Thus, it can be guaranteed that the representational power is unrestricted.

4. EXPERIMENTAL ANALYSIS

To investigate the performance of the IHRTM we employ real world data from the eBay example used for illustration in the previous section. Before the empirical results of our experiments will be presented, we first describe the experimental setup. In the following sections three different aspects of the IHRTM's performance were investigated in the course of our research: First the algorithm's abilities to characterize a trust-situation by clustering are investigated in Section 4.2. Second the predictive performance concerning trust values is tested. Finally, the learning efficiency is analyzed in the context of dynamic behavior of non-stationary trustees. As the later cannot be analyzed within the eBay scenario we used interactions recorded from a negotiation game. The experimental setup and evaluation is covered in Section 4.4.

4.1 Experimental Setup: eBay-User Ratings

eBay feedback-profiles are a valuable source of easily accessible data that expresses human-trust assessment. Every eBay member has a public feedback profile where all items he has bought or sold in the last 90 days are listed with the associated feedback ratings he received. In addition the feedback profile includes statistics on all transactions of the user.

We gathered data from 47 sellers that on the one hand had at least 10 negative or neutral ratings and on the other hand sold items in at least one of 4 selected categories from the lowest level within the eBay-taxonomy. The former is important because negative or neutral user-ratings on eBay are rather rare. To further balance the ratio of positive vs. negative/neutral ratings we only evaluated as many positive rated transactions as there were negative/neutral ones. In this way the data-set is stratified, meaning that there is an equal number of positive and negative ratings per seller.

Attributes Att^a of the seller were directly extracted from the feedback profile. We picked the positive feedback and the feedback score and discretized both in 2 and 5 classes, respectively. For the item attributes Att^s we chose the top level category in the eBay taxonomy on the one hand, resulting in 47 discrete states. On the other hand we collected the item condition which is a binary feature: either new or used.

From those 47 hand-picked sellers we gathered a total of 1818 rated sales of 630 different items. Two items were assumed to be alike if they were in the same lowest level category and their attributes were identical. Relation attributes are always of size $seller \times items$, so Att^c and Att^t both are sparse matrices with 47×630 possible entries. The non-zero entries indicate that this seller has sold this item.

As we wanted to keep the computational complexity low we only considered binary relational attributes Att^c resp. Att^t. For Att^c we chose the binarized final price of the auction and for Att^t the rating. Negative and neutral ratings were both treated as negatives.

After having extracted the data, the GS-process to train the IHRTM was started. In the beginning the sellers and items are re-clustered intensely and both cluster assignments and cluster sizes are unstable. Once the Markov chain starts to converge the cluster sizes tend to stabilize and eventually, the training can be stopped. The decrease of the cluster sizes is exemplarily shown in figure 3 for one cross-validation run.

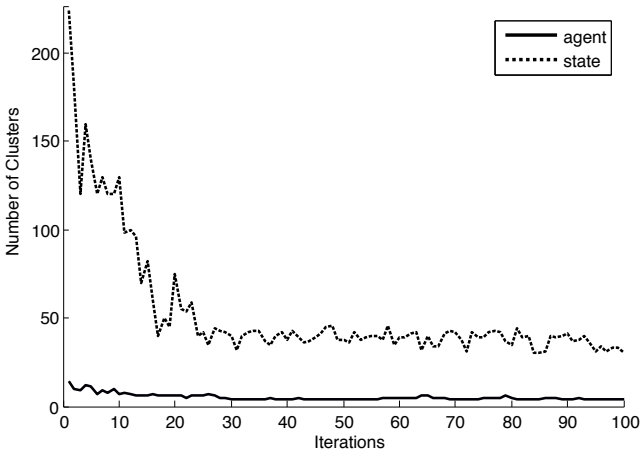

Figure 3: Trace of the number of agent- and state-clusters up to 100 iterations.

4.2 Clustering Effect

After the clusters have stabilized we can visualize two interesting facts about the trust situation.

First, we can plot a matrix showing the assignments of each seller to a cluster. This can provide knowledge about how many different clusters exist, which are the most popular clusters and which elements are grouped together. After convergence, the 47 sellers were assigned to 4 clusters as shown on the left half of figure 4. The same assignment matrix can be generated for the items clsuter assignment but since there are 613 items and 40 item clusters, we did not plot the matrix and simply show its symbol Z^s on top of the right matrix in figure 4.

Second, the posterior probability $P(Att^c, Att^t|Z^a, Z^s)$ can be visualized. The matrix on the right side in figure 4 illustrates the probability of getting a positive rating given the cluster assignments of a seller and a item. A darker value indicates a higher probability of being trustworthy in a given interaction. Now, picking a row (representing an agent cluster) or a column (representing a state cluster) we can identify clusters that are in general more trustworthy than others.

4.3 Predictive Performance

In order to judge the performance of predicting the trust value Att^t we compared the results of IHRTM with two other standard machine learning algorithms, namely a support vector machine (SVM) using a PolyKernel and a Decision Tree ($DecTree$) both from the Weka toolbox [11]. Since those algorithms are both propositional learners, meaning they cannot handle a relational data representation but only a vector of independent and identically distributed features plus a label, we had to "flatten" the data first. By transforming the data into a flat representation, also known as "propositionalization", the structural information can get lost. In general there is no standard propositionalization procedure (see [5]). The potential low quality of propositional features is not crucial in our simple scenario but becomes increasingly problematic in more complex relational models.

We propositionalized the data in three different ways: First, we only considered the target trust variable Att^t and tried to predict trustworthiness by the mere rate of positive feedback as it is done in most existing statistical trust models

	Accuracy	ROC Area
Ratio	48.5334 (±3.2407)	-
SVM	54.1689 (±3.5047)	0.512 (±0.0372)
DecTree	54.6804 (±5.3826)	0.539 (±0.0502)
SVM+ID	56.1998 (±3.5671)	0.5610 (±0.0362)
DecTree+ID	60.7901 (±4.9936)	0.6066 (±0.0473)
IHRTM	**71.4196** (±5.5063)	**0.7996** (±0.0526)

Table 1: Predictive performance on eBay-user ratings

(see *Ratio* in table 1). Clearly, the result cannot be better than random guessing as the data-set is stratified. However, this demonstrates that the assumption of context independency made by many trust models is fatal when trust observations are uniformly distributed. Second, we tested the performance of the propositional algorithms with all features - namely Att^a, Att^s, Att^c and again Att^t - as the label. As a result we extracted 1818 samples with 5 features and one label, each. This way, the same features are available to the propositional learners as they are to the IHRTM. Third, we accounted for the missing relational information (which seller sold which product) by introducing two further features: An ID-number for the seller and the item, respectively. So the input to the propositional learners was a 1818×8 matrix in this setup.

The result of all 3 setups is shown in table 1. We report the accuracy of predicting positive ratings as well as the AUC (also called ROC area). This measure represents the area under the *receiver operating characteristic curve* which is used for evaluating binary classifiers that can output probabilities instead of binary decisions. In all our experiments we averaged our results using 5-fold cross-validation. The accompanying 95%-confidence intervals are reported as well. Finally the prediction performance is also evaluated for the IHRTM and compared to the previous attempts (see table 1).

In general, the task of predicting eBay-user ratings seems to be difficult, which can be explained when reading the comments assigned to the ratings. The reasons for a positive or a negative evaluation are most of the time not related to specific properties of sellers or items but a unique incident. Besides that, the high incentives to give positive ratings despite having had negative experience are a general and well known flaw in the eBay-feedback mechanism: sellers usually wait for the buyer's rating before they rate the buyer. Thus, buyers often give positive rating just to receive a positive rating from the seller as well. As a response to this problem, eBay is planing to introduce a new feedback mechanism in May 2008.

Still, the IHRTM's performance clearly outperforms random guessing and could verifiably outperform the propositional learners. This is most likely due to the collaborative filtering effect, that can only be utilized by the IHRTM. Thus, there seems to be a gain if learning with the assumption that e.g. when two sellers sell similar items they might be comparable in their trust-ratings. More precisely, if two sellers both got positive ratings after selling one specific item their ratings might be comparable when selling a different item as well. Or the other way round, if two items both got positive ratings after sold by one specific seller their ratings might be comparable when sold by a different seller as well.

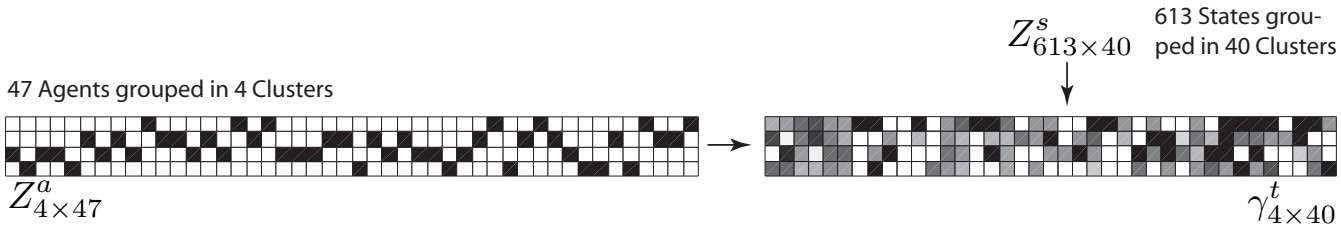

Figure 4: Left: Final clustering of trustees Z^a. Top right: Items Z^s. Bottom right: $P(\gamma^t|Z^a, Z^s)$

4.4 Learning Efficiency

As mentioned in the introduction, the learning efficiency[1] and the ability to rapidly adapt is crucial, especially in so called initial-trust situations or in situations where the trustee does learn and adapt as well. To evaluate the performance concerning learning efficiency, we had to use a different, more controlled experimental setup as in the previous eBay example. Only if we know about the stationarity of agents we can compare the performance of an adapting agent to a stationary agent. For this purpose, we recorded interactions in a simulated strategic two-player negotiation scenario.

4.4.1 Experimental Setup: Negotiation game

Finding an agreement amongst a group of conflicting interests is one of the core issues of distributed artificial intelligence. Auctions, information markets, preference and judgement aggregation, game theory and automated negotiations are all research areas that deal with those kind of problems. However, most of the approaches neglect the fact that finding the best agreeable solution is not sufficient if commitments can not be enforced by the interaction mechanism or the incentives of the opponents can not be inferred. In order to investigate this issue we extended the implementation of a multiagent trading framework by an additional negotiation step.

In the chosen scenario, players try to collect a certain number of resources for selling. Only one type of resource can be collected at a time. In each round, every player gets new random resources from the deck and some resources must be added to the stack of collected resources. If the types of the resources previously held in the stack and the types of new resources are not identical, all resources collected so far are wasted. To avoid this, players can trade with other players and exchange some of their resources. Resources received from fellow players are pushed onto the stack.

As defined before, let c be the commitments that the agents are negotiating over. The outcome of this negotiation is specified by a set of binary features Att^c. Now, given a set of commitments c that two agents have agreed on and promised to fulfill, the agents enter an additional trading step in which each of them is free to decide which action to take. This way, the agent can decide whether to stick to a commitment or break it at will.

One interaction-round consists of three phases:

Negotiation: Each agent a follows a predefined strategy that proposes a potential set of actions c both parties might agree on (e.g., an exchange of goods). In doing so, agents have neither knowledge of the actions available to the other agents nor their reward function. Thus, agents can propose an infeasible action to convince the opponent to act more to their favor. Received proposals can be rejected and counter-proposals can be made resulting in an iterative approximation of a solution. The negotiated result is considered as a commitment attribute Att^c.

Trading: This is the final decision made by every agent whether to stick to a commitment or break it. Finally, the action t chosen by agent a is executed accordingly.

Evaluation: The agents can review the effective actions t of the opponent by observing the received goods and draw conclusions for future interactions. The next stage game is sampled according to a stochastic transition function.

This procedure is repeated over a potential infinite number of rounds with different types of agents playing against each other.

4.4.2 Evaluation

Three different agent types with two different negotiation strategies and three different trading strategies were used as opponents in the negotiation game.

The two negotiation strategies are both stationary and are based on a monotonic concession protocol (cf. [2]). The agents denoted *Honest* and *Fictitious* only propose actions that they actually could perform, while agent *Greedy* also offers and accepts infeasible actions with the intend to achieve an opponent action with higher payoffs. Both strategies iteratively propose a larger set of actions by lowering their expected utility and offering less favorable outcomes.

Each agent type plays a different trading strategy where *Honest* and *Greedy* are stationary and *Fictitious* is adaptive. *Greedy* always maximizes its utility regardless of c, while *Honest*-agent always sticks to c. At last, *Fictitious* plays according to the *fictitious play* algorithm. It's a traditional learning algorithm from game theory for repeated games, where the opponent is assumed to be playing a stationary strategy. The opponent's past actions are observed, a mixed strategy is calculated according to the frequency of each action and then the best response is played, accordingly.

In every round that was played the commitment c and the effective outcome t were recorded and features Att^s, Att^c and Att^t were extracted. No specific attributes for Att^a were available except for the identity of the agent. Three discrete features Att^s from s where calculated describing the average payoff over all possible opponent actions, the maximum

[1]By *learning efficiency* we do not mean computational complexity of the learning algorithm, but numbers of observations needed to make effective predictions.

possible payoff and the number of feasible actions. Att^c describes a single binary feature stating whether there is a feasible action that could be carried out and would result in a positive reward if the negotiated commitment was carried out by the opponent. The same feature was recorded for Att^t after the actual action took place.

In this way a total of 600 interactions, 200 per agent type, containing a total of 289 different stage games were recorded. The input for the IHRTM consisted of three Att^s vectors with 289 elements, and two 289×3 matrices for Att^c and Att^t. Again, for a comparison with propositional machine learning algorithms the data was propositionalized, resulting in 600 feature vectors with $3 \times Att^s + 1 \times Att^c$ elements and 600 corresponding labels. As before, the content based algorithms were also evaluated with an agent- and state-ID as an additional feature. The evaluation procedure is the same as in the eBay experiments.

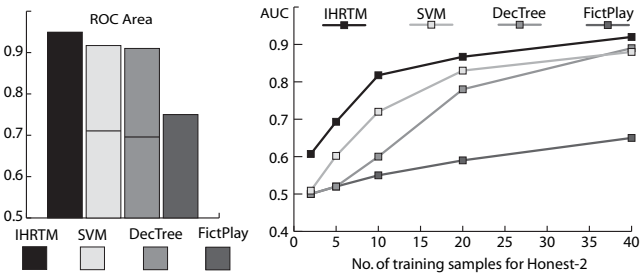

Figure 5: Results for play against $Honest$. Bar graph on the left: AUC for classifying Att^t. Graph on the right: learning curve for increasing number of training data for the additional *Honest-2*.

The overall performance according to AUC is depicted in the bar graph on the left of Figure 5. IHRTM shows a slightly better performance in classifying Att^t than SVM and DecTree. Without the agent-ID as an additional feature the performance of DecTree and SVM drops considerably (black line at around 0.7). Again, we explain the superior performance by IHRTM's ability to exploite cross-entity dependencies. $Fictitious$, as expected, performs much worse as it is not able to generalize over different interactions and can't make use of the context provided by Att^s and Att^c.

The inherent clustering ability of IHRTM suggests that it is especially well suited for rapid adaptation when unknown but related agents and conditions are observed. Actually, entities can be correctly assigned to a cluster without having seen a single effective Att^t related to this entity just by the other attributes. To check this assumption we gathered data from interactions with another $Honest$ type agent and evaluated the performance for different numbers of training samples. On the right of Figure 5 the learning rates for agent *Honest-2* are plotted. The results confirm that especially for a small sample size ≤ 20 the performance of IHRTM is clearly better compared to the content based approaches.

In contrast, the performance in the task of trying to predict $Fictitious$ is clearly worse for all of the techniques (see Figure 6). Expectedly, IHRTM, SVM and DecTree cannot handle dynamic opponents. Again, the IHRTM is most competitive in terms of efficient learning.

In addition, the IHRTM offers another advantage over the other techniques. The predictions are based on an inher-

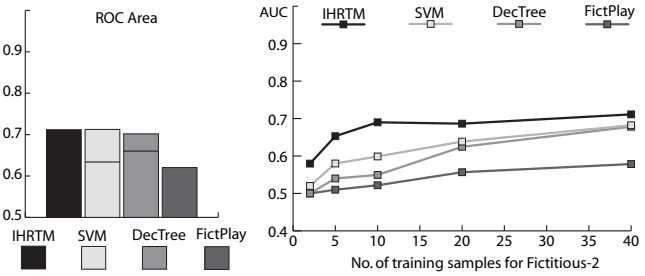

Figure 6: Results for play against $Fictitious$. Bar graph on the left: AUC for classifying Att^t. Graph on the right: learning curve for increasing number of training data for the additional *Fictitious-2*.

ent construction of clusters of Z^a and Z^s. The fast learning rate indicates that a previously unknown trustee is correctly assigned to an existing cluster if this type of agent has been observed before. Consequently, once *Fictitious-2* is assigned to the "$Fictitious$-cluster" IHRTM could assess its performance on this cluster and eventually suggest a different learning scheme for agents in this cluster. In other words it can identify non-stationary behaving agents.

Figure 7 visualizes the final cluster sizes and cluster assignments. The top right matrix shows the assignment of seven different agents to Z^a. All three agent types were clustered correctly into three groups (columns). To evaluate this further we generated data from another stationary opponent with a different trading strategy that is very similar to $Honest$: $Sneaky$-agent only deviates from c if it can increases its utility by a large margin. Interestingly, the assignment of $Sneaky$- and $Honest$-agent to the same cluster suggests that this strategy might effectively build trust.The matrix in the lower left corner of Figure 7 visualize Z^s. From 289 stage games (columns) 8 different clusters (rows) emerged. This is an impressive reduction in complexity while still having good classification results. The two stacked matrices in the bottom right corner represent Att^t and Att^c (below). Each row indicates one state cluster, each column an agent cluster. Brighter rectangles indicate a lower probability for a positive reward. As expected, the first column ($Greedy$ cluster) is on average brighter than the second and third column ($Honest$ and $Fictitious$ cluster). All those observations, including the misclassification of $Sneaky$, correspond well to human intuition.

5. RELATED WORK

As already pointed out, connecting trust to the trusted agent alone without considering contextual and other aspects (dimensions) of trust is not sufficient in many scenarios. Whereas much research on trust concedes the importance of context information, most of them do not actually use such information for the calculation of trust degrees in a general and automated way [9]. So far, only one approach also models context by taking into account identity and state (see [6]). Besides that, using contextual information for initial trust assessment and the transfer of trust between contexts is novel to our knowledge.

Analogously, we argue that a fine grained modeling of relations between agents and their environment is essential to capture the essence of trust, especially in initial trust situa-

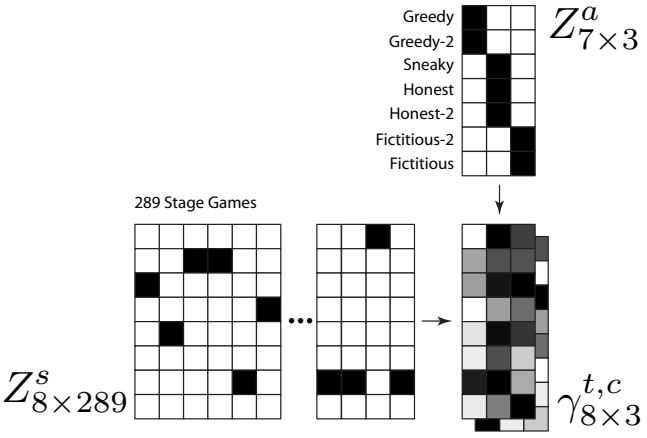

Figure 7: Final clustering of agent types and states (Z^s and Z^a). Bottom right: $P(\gamma^t, \gamma^c | Z^s, Z^a)$

tions. There exist a few approaches that can take relationships into account when modeling trust. But in most of this research such relationships are either only considered as reputation or recommendations [7], or as interactions between a group of agents (e.g., [1]). The diverse kinds of relations that exist between two agents in a specific situational context are not modeled in detail. In addition, most learning techniques are optimized for one specific scenario only.

Assessing initial trust values for unknown agents based on pre-specified membership to a certain group has been addressed by [8]. A group-based reputation architecture is proposed here where new agents are assessed according to their pre-specified membership to a certain group of agents. Likewise, the *TRAVOS-C* system proposed by [9] includes rudimentary ideas from hierarchical Bayes modeling by assigning parameter distributions to groups of agents but doesn't come to the point to give a fully automated and intuitive way of how to learn infinite hidden variables.

6. CONCLUSIONS AND FUTURE WORK

In this contribution, we presented a context-dependent way to build statistical relational trust models in general and our *Infinite Hidden Relational Trust Model* (IHRTM) in particular. We demonstrated how trust can be modeled and learned in theory and in two experimental setups: first, a real world data set from the *eBay feedback-system* and second a simulated negotiation game.

Our experimental results suggest that the IHRTM offers advantages in 3 different dimensions. First, the inherent clustering capabilities increase interpretability of trust situations. Second, the predictive performance can be improved compared to a "flat", feature-based machine learning approach if trained with relational data that exhibit cross-attribute and cross-entity dependencies. Third, the IHRTM is especially well suited for rapid adaptation because of its ability to transfer knowledge between related contexts.

While the IHRTM cannot handle trustees with strategies that are non-stationary effectively, it can identify non-stationary agents. An adaptive learning strategy could be part of future work. Furthermore, we plan to extend our framework to scenarios with arbitrary numbers of concurrently interacting trustees. the same time. While propositional machine learning algorithms cannot be easily applied in this case it can be realized by relational models. Furthermore, we are currently comparing the complexity and performance of different inference algorithms.

We introduced statistical relational trust learning in general and presented the IHRTM in particular. The goal of our work is to advance research on computational trust by making modeling and learning of trust more applicable, efficient, intuitive and interpretable.

7. REFERENCES

[1] R. Ashri, S. D. Ramchurn, J. Sabater, M. Luck, and N. R. Jennings. Trust evaluation through relationship analysis. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 1005–1011, New York, NY, USA, 2005. ACM Press.

[2] U. Endriss. Monotonic concession protocols for multilateral negotiation. In *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 392–399, New York, NY, USA, 2006. ACM Press.

[3] D. Heckerman, C. Meek, and D. Koller. Probabilistic models for relational data. Technical Report MSR-TR-2004-30, Microsoft Research, 2004.

[4] C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. In *AAAI*, 2006.

[5] M.-A. Krogel. *On Propositionalization for Knowledge Discovery in Relational Databases*. PhD thesis, die Fakultät für Informatik der Otto-von-Guericke-Universität Magdeburg, 2005.

[6] M. Rehak and M. Pechoucek. Trust modeling with context representation and generalized identities. In M. Klusch, K. V. Hindriks, M. P. Papazoglou, and L. Sterling, editors, *CIA*, volume 4676 of *Lecture Notes in Computer Science*, pages 298–312. Springer, 2007.

[7] J. Sabater and C. Sierra. REGRET: reputation in gregarious societies. In J. P. Müller, E. Andre, S. Sen, and C. Frasson, editors, *Proceedings of the Fifth International Conference on Autonomous Agents*, pages 194–195, Montreal, Canada, 2001. ACM Press.

[8] L. Sun, L. Jiao, Y. Wang, S. Cheng, and W. Wang. An adaptive group-based reputation system in peer-to-peer networks. In X. Deng and Y. Ye, editors, *WINE*, volume 3828 of *Lecture Notes in Computer Science*, pages 651–659. Springer, 2005.

[9] W. T. L. Teacy. *Agent-Based Trust and Reputation in the Context of Inaccurate Information Sources*. PhD thesis, Electronics and Computer Science, University of Southampton, 2006.

[10] V. Tresp. Dirichlet processes and nonparametric bayesian modelling. Tutorial at the Machine Learning Summer School 2006 in Canberra, Australia, 2006.

[11] I. H. Witten and E. Frank. Data mining: practical machine learning tools and techniques with java implementations. *SIGMOD Rec.*, 31(1):76–77, 2002.

[12] Z. Xu, V. Tresp, K. Yu, and H.-P. Kriegel. Infinite hidden relational models. In *Proceedings of the 22nd International Conference on Uncertainity in Artificial Intelligence (UAI 2006)*, 2006.