Representing Context for Multiagent Trust Modeling

Martin Rehák, Miloš Gregor, Michal Pěchouček
Department of Cybernetics
Czech Technical University in Prague
Technická 2, Prague, 166 27, Czech Republic
{mrehak,pechouc}@labe.felk.cvut.cz

Jeffrey M. Bradshaw
Institute for Human and Machine Cognition
40 South Alcaniz Street
Pensacola, FL, USA
jbradshaw@ihmc.us

Abstract

We present a universal mechanism that can be combined with existing trust models to extend their capabilities towards efficient modelling of the situational (context-dependent) trust. The mechanism describes the similarity between the situations using their distance in a metric space and defines a set of reference contexts in this space to which it associates the trustfulness data. The data associated with each reference context is updated and queried with the weight that decreases with distance between the current situation and the reference context. In the presented mechanism, we use Leader-Follower clustering to place the reference contexts to be representative of the data. In an empirical test, we show that context-aware models easily outperform the general trust when the situation has an impact on partner trustfulness and that their performance and efficiency is comparable with general trust models when the trustfulness is independent of the situation. Multi-context nature of the model also expands its use towards more advanced uses, allowing policy/norm learning from at the trust model at runtime, as well as reasoning based on uncertain identities.

1 Introduction

While the need to consider the situation in the trusting decision has been recognized for a long time [10], this issue has been long neglected by the trust research community. Sabater & Sierra ([15], page 39) have recently even stated that: *"...there are very few computational trust and reputation models that care about the multi-context nature of trust and even fewer that propose some kind of solution..."*.

This contribution presents a mechanism that converts a general (denoted single-context in [15]) trust model into a situational one. The mechanism uses metric spaces to exploit the similarity between the situations, addressing two critical properties of any such model: data efficiency and relevance:

◇ **Data Efficiency** means that the learning phase of the model shall be relatively fast, exploiting the similar data to infer the conclusions about the situations that were not previously encountered. It also ensures that the model consumes a reasonable amount of memory and processor time.

◇ **Data Relevance** means that the conclusion we infer about the situation shall be based on the most relevant experience, not on the general behavior of the partner.

The two properties mentioned are contradictory in their nature (ignoring the implementation and mechanism details) : with the increasing relevance, we sacrifice the efficiency as we need more data to discover the details. Therefore, the mechanism we seek shall be adaptive, able to discover and better represent the important situations.

We shall note that the mechanism as presented is not a trust model by its own right – it is an extension that can be combined with most existing trust models and we prove this ability in the evaluation section, where we test it with two different trust models. In this paper, we will distinguish between the situation and the context: **situation** is the state of the reality in the moment of the trusting decision or observation; the **context** is a formal, simplified representation of the situation in our formal model shown in Section 2.

In Section 2, we introduce our model and explain its functioning, including two alternative approaches to dataset definition in Section 2.3. In Section 3, we evaluate the model empirically and determine its performance under various conditions and with two different trust models. In Section 4, we present the model extensions towards the inductive reasoning and prediction, before concluding in Section 5 where we also present our future work.

2 Formal Model for Context Representation

In order to represent the situation of the trusting decision, we replace the situation by its context, a point c_i in the context space $\mathbb{C}$. Each dimension of the Q-dimensional metric space $\mathbb{C}$ corresponds to one relevant feature of the

situation, and the metrics $d(c_1, c_2)$ defined on $\mathbb{C}$ describes the similarity[1] between the contexts c_1 and c_2.

2.1 Context Space Definition

In general, we define the metric space $\mathbb{C}$ in several steps: (*i*) We identify all relevant features of the environment, then (*ii*) define the Q-dimensional context space where each dimension q matches a relevant feature. (*iii*) For each dimension q, define its quantification (either discrete or continuous) and appropriate distance metric d^q that correctly represents the feature, and finally (*iv*), we define a joint metric d on the full space $\mathbb{C}$, taking into considerations the domain characteristics and marginal metrics d^i[2].

To combine the marginal distances into the d function, we will typically choose one of the special types of Minkowski distance:$d(c_1, c_2) = (\sum_{q=1}^{Q} |c_1^q - c_2^q|^p)^{\frac{1}{p}}$. For many practical purposes, we choose the values of p to be 1, defining so called Manhattan distance that adds the marginal distances of each dimension, or 2 to define an Euclidean distance, or we pose $p \to \infty$, obtaining Chebyshev distance defined as a maximum of marginal distances.

2.2 Reference Contexts and Trustfulness Values

Once we have defined the metric space $\mathbb{C}$ with its distance function d (see Section 3 for an example), we have a formal framework how to asses the similarity of two trusting situations. In order to integrate the context representation framework with the trust model, we need to define a set $\mathcal{R}$ of *reference contexts* r_i, the points in the $\mathbb{C}$ for which we keep the trustfulness values. We shall note that while the $\mathbb{C}$ definition shall be the same for all partners modeled by the trusting agent, the set $\mathcal{R}$ and the associated trust values are kept independently for each partner.

In practice, this means that instead of maintaining a single instance of the trust model structure (representing general trust) per partner, we shall maintain one instance per partner for each relevant reference context. The metrics d is used to determine the weights of individual reference contexts in the evaluation or observation of a specific trusting situation. In the following, we will denote as $\Theta_A(X|r_i)$ the trustfulness of agent A in the situation represented by reference context r_i. This value can be updated according to most trust models currently in use, for example Regret [14], FIRE [9] or other [12, 6, 13], provided that the inputs to this model can be weighted and the outputs aggregated.

To obtain the weights of individual reference contexts r_i for updates or aggregation after the event/query described by context c_d, we transform the distance $d(c_d, r_i)$ as follows: $w_i = f(d(c_d, r_i))$, where f is a non-increasing function on $[0, +\infty)$. This function represents the decay of the observation usefulness with increasing distance d of the particular reference context r_i – obviously, it is most useful when its distance $d(c_d, r_i)$ from the reference context is zero. This function, together with the metric, is a part of the domain description. For example, in our experiments presented in Section 3 we use a simple form of weight function defined as $w_i = e^{-d(c_d, r_i)}$.

After each **observation** we integrate the new observation $\tau_A(X|c_o)$ into the apriori trustfulness evaluation $\Theta_A^p(X|r_i)$ (where p is the number of previous observations, with aggregate weight $W^p = \sum_{j<=p} w_i^p$) for each r_i (where w_i is non-zero) using the weighted aggregation formula:

$$\Theta_A^{p+1}(X|r_i) = WeAg((\Theta_A^p(X|r_i), W^p, (\tau_A(X|c_o), w_i^{p+1})) \quad (5)$$

The exact form of the $WeAg()$ operator depends entirely on the trust model used to represent $\Theta_A^{p+1}(X|r_i)$. In the trivial case, when the $\Theta_A^p(X|r_i)$ is just a w_i weighted average of all p previous observations, we obtain:

$$\Theta_A^{p+1}(X|r_i) = \frac{W^p \cdot \Theta_A^p(X|r_i) + w_i^{p+1} \cdot \tau_A(X|c_o)}{W^p + w_i^{p+1}} \quad (6)$$

When we **query** the model to take a trusting decision, the current context c_d is determined and the trustfulness is obtained as a weighted combination of trustfulness associated with respective reference contexts.

$$\Theta_A(X|c_d) = WeAg_{r_i \in \mathcal{R}}(\Theta_A(X|r_i), w_i) \quad (7)$$

In the weighted average case, we obtain:

$$\Theta_A(X|c_d) = \frac{\sum_{r_i \in \mathcal{R}} w_i \cdot \Theta_A(X|r_i)}{\sum_{r_i \in \mathcal{R}} w_i} \quad (8)$$

2.3 Reference Set Shapes

As we have seen in Equations 5 to 8, the computational and memory complexity of trust processing depends linearly on the size of the set $\mathcal{R}$. Therefore, it is crucial to keep the size of the set as small as possible, while ensuring that the bulk of the data is well covered by the contexts in their proximity. When suggesting a method for shaping

[1] Any distance function $d : \mathbb{C} \times \mathbb{C} \to R$ must respect following properties: **non-negativity**:

$$d(c_1, c_2) \geq 0 \quad (1)$$

, **symmetry**:

$$d(c_1, c_2) = d(c_2, c_1) \quad (2)$$

, **zero distance $\Leftrightarrow$ identity**:

$$d(c_1, c_2) = 0 \Leftrightarrow c_1 = c_2 \quad (3)$$

, **triangle inequality**:

$$d(c_1, c_3) \leq d(c_1, c_2) + d(c_2, c_3) \quad (4)$$

.

[2] Alternatively, we may define the global metric directly, without marginal ones

```
class LFClustering:
  def __init__(self, threshold):
    self.centers = []
    self.thresh = 0.5
  def newSample(self, sample):
    closest,dist = self.findClosestCenter(sample)
    if closest != '0' and dist <= self.thresh:
      closest.aggregate(sample, 1, 0.01)
    else:
      self.centers.append(sample)
  def findClosestCenter(self, sample): ...
```

Figure 1. L-F Clustering algorithm [8] outlined in Python

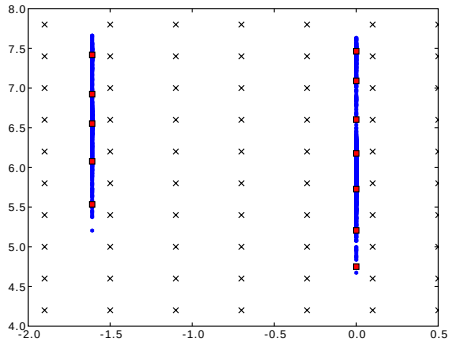

Figure 2. Regular grid (crosses) and clustering-based reference contexts (red squares) covering the data (blue dots) in a 2D $\mathbb{C}$ projection.

the set $\mathcal{R}$, we shall avoid the introduction of a big number of parameters to assign, as the domain-dependent metrics shall shield us from the tuning process. In both methods we present, only a single distance parameter is required.

As a baseline approach for our experiments, we have used a regular grid form for the $\mathcal{R}$ definition. In this approach, the reference contexts are generated as a grid, spaced regularly in each dimension by grid distance parameter. When in the 3D, the resulting structure is similar to a simple cubic crystal lattice and covers the space $\mathbb{C}$ (or rather its relevant area) regularly. In Fig. 2, we can see such grid as a regular lattice of black crosses over the space $\mathbb{C}$.

The regular grid approach is obviously an inefficient one, as in the real applications the observations are rarely spread uniformly over the space. They rather tend to form clusters of points in the $\mathbb{C}$ representing the typical trusting decisions. This motivates our principal approach, where we use the pattern matching techniques to define (and also update the positions) of the reference contexts r_i.

From the wide range of clustering techniques, we have selected the Leader-Follower clustering [8], that doesn't need any information regarding the expected number of clusters (see Fig. 1). Instead, it uses a cutoff distance parameter similar to grid distance – when the new observation falls farther away from an existing cluster than the cutoff distance, a new cluster is created around this observation. Position update of the existing centroids (e.g. reference contexts in our case) is analogous to other clustering methods. This corresponds very well with the real world effects, e.g. inflation or seasonal variations.

3 Evaluation

To evaluate our approach experimentally, we model the trust reasoning of a humanitarian aid organization agent that acquires transportation services from several local transporters after major disaster.

Context Space Definition To illustrate the abstract notions of metric space $\mathbb{C}$, we introduce an example of such space for our logistics scenario, where we model each trusting situation (observation or decision) by three parameters: cargo type, cargo size and road quality. Cargo *type* defines the product we transport: medical supplies, food or durable goods. Each cargo type has specific handling requirements – medical supplies are the most sensitive to carry, while the durables require less care. *Size* of the transport is simply a quantity to carry, while the *road quality* represents the quality of the roads to use for transport. It is interesting to note that *type* dimension is discrete, while the *size* and *road quality* are real-valued, but different: one has an absolute scale (size), while the other will be close to 1.

The context space $\mathbb{C}$ is three dimensional, with one discrete dimension and two continuous ones. The next step is a definition of marginal distances d^q for each dimension. In the *type* domain, we place our products on a "sensitivity" scale: medical supplies require most attention: 5, with the food in the middle: 1 and the durables as least sensitive ones, with 0.2 value[3]. Our type distance metrics is defined as follows, using the product properties defined above:

$$d^{type}(c_1, c_2) = |ln(type_1) - ln(type_2)| \qquad (9)$$

In the size domain, the metric shall describe the similarity between two contracts in terms of their relative size. We propose a measure

$$d^{size}(c_1, c_2) = |ln(size_1) - ln(size_2)| \qquad (10)$$

[3]Inverting the scale will not change the result thanks to the distance symmetry stated in Eq. 2.

The logarithmic relation captures an intuitive notion of ratio: 10 tons difference between two 20 and 30 ton transports is much more important than the same difference between two shipments of thousands of tons.

We apply the same reasoning for the road quality:

$$d^{road}(c_1, c_2) = |ln(qual_1) - ln(qual_2)| \qquad (11)$$

Then we combine the above metrics using a slightly modified (weighted) "Manhattan distance":

$$d(c_1, c_2) = \alpha_1 d^{type}(c_1, c_2) + \alpha_2 d^{size}(c_1, c_2) + \alpha_3 d^{road}(c_1, c_2) \qquad (12)$$

Experimental Setup In the task allocation problem that the agents solve in the simulated humanitarian logistics scenario, the agents choose one or more providers (transporters) for each contract and use their trust models to reason about their trustfulness.

In the underlying simulation model, the transporters answer the call for proposals with *bid prices* pr_b based on the nominal transportation cost and profit margins. The *real price*, that includes the cost of the cargo lost during transportation, is derived after the transport from the bid price and transporter *real trustworthiness* Θ. The Θ depends on the same parameters as those that define the $\mathbb{C}$ dimensions. Real price pr_r is determined as $pr_r = \frac{pr_b}{\Theta}$, where

$$\Theta = \Theta_{type} \cdot atan'(price) \cdot atan'(supply) \qquad (13)$$

The function $atan'(x)$, used as a sigmoid approximation, is defined as a normalized $arctan$: its range is (x_{inf}, x_{sup}) (both x_{inf}, x_{sup} are in the range set) and x coordinate of its flection point is defined by parameter x_{center}. x_{slope} determines the first derivation - speed of the growth on the domain.

$$atan'(x) = \frac{1 - x_{inf}}{\pi} \cdot arctan(\frac{x_{center} - x}{x_{slope}}) \qquad (14)$$

While the provider simulation is a very simple one, it is sufficiently versatile to model the performance of market actors to obtain validation scenarios for our methods.

To evaluate the performance of the evaluated trust models, we introduce the *mean loss*, defined as a difference between the real price pr_r and the bid price pr_b. In the graphs, it is aggregated per all contracts awarded in a single time step. As it is impossible to achieve the zero loss in our scenario, we introduce the optimal choice value, defining the optimal performance of the trust model.

To validate the model independence of the method presented (i.e. the fact that the restrictions placed on Θ modelling are not constraining), we have used two different trust models in our evaluation. The first model, denoted **RNT** in the graphs, we represent the trustfulness in each $\Theta(X|r_i)$ as a time-weighted average of the last N relevant observations. This means that we store N real values in each reference context r_i. The other model, denoted FNT, is a slightly simplified representation described in [13], where each $\Theta(X|r_i)$ is a triangular, asymmetrical fuzzy number.

In Figures 3, 4, 6 and 8, we compare the performance of the trust models without the context representation (denoted **FNT** and **RNT**) with the same models enhanced with context representation. The first pair, denoted **grid RNT** and **grid FNT** uses the grid form of the reference set $\mathcal{R}$. The models **clusters RNT** and **clusters FNT** use the form based on the leader follower clustering.

3.1 Influence of Situation Modelling

In the first batch of experiments, we will investigate the influence of the context modelling. We will therefore compare several trust models with and without the context components in the scenarios with increasing level of situation influence on the provider performance. The changes in the performance are modelled by changes of the coefficients in the Eq. 13 and 14.

In the first scenario (Fig. 3), the performance of all the providers is flat over the whole space $\mathbb{C}$ - the outcome of the delegation/contracting is independent of the situation. We may note that the general methods perform slightly better, as their learning process is more efficient, but the differences remain minor.

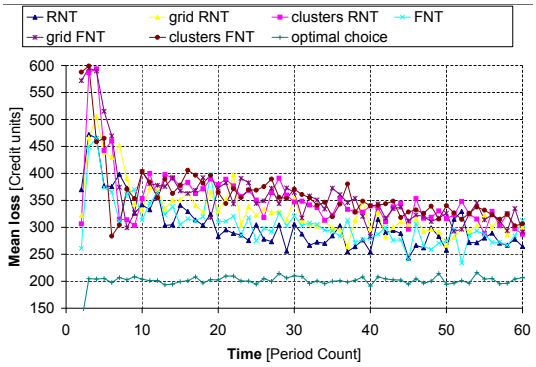

Figure 3. Scenario with trustfulness independent of situation – all methods perform comparably.

In the second scenario (Fig. 4), we have introduced a strong, but one dimensional situation dependence with one best provider per cargo type. We can see that in this case, context-based methods easily outperform the general trust and reach the optimum relatively fast. In Fig. 5, we can see that the depicted sub-market (defined by the contracts

in one part of the context space) is rapidly dominated by the most trustful provider, while the others are restrained to the services where they perform better.

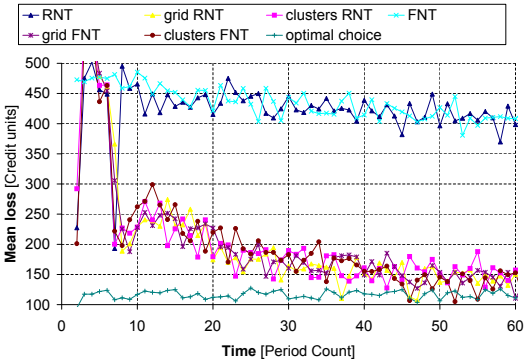

Figure 4. Scenario with trustfulness dependent on the cargo type only.

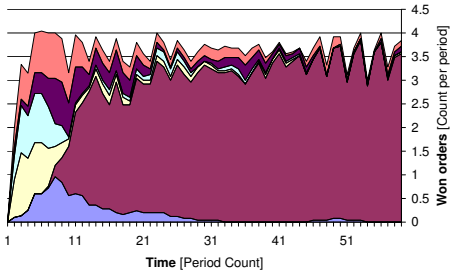

Figure 5. Market shares example with trustfulness dependent on the cargo type only.

When we introduce a full 3D context dependence, we obtain the results shown in Fig. 6 and Fig. 7. We can see that the task is more difficult due to the increased dimensionality, but the context modelling solves the problem. The slower learning pace is clear when we compare the Fig. 7 with Fig. 5 – the market domination is slower. In Fig. 6, se shall note that the clustering based metrics provide better results than the grid based ones. We attribute this difference to the fact that the reference context points cover the data better, as shown for example in Fig. 2.

3.2 Metrics Quality

The definition of the distance function d and weight function w_i as discussed in Section 3 is crucial for the trust modelling quality. In Fig. 6 and Fig. 8, we can compare the influence of the inappropriate metrics (results shown in Fig. 8) when compared with an appropriate metrics in Fig. 6. The difference between the two cases is merely in

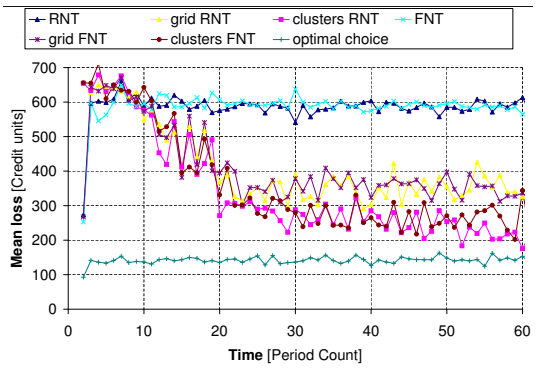

Figure 6. Scenario with trustfulness dependent on all 3 parameters, with an adequate metrics. Compare with Fig. 8.

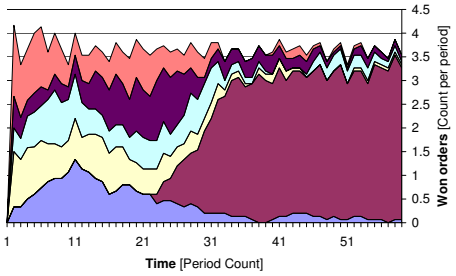

Figure 7. Market shares with trustfulness dependent on all 3 parameters. Compare with Fig. 5.

the values of the $\alpha_{1..3}$ coefficients from Eq. 12. In the case depicted in Fig. 8, the values are $\alpha_1 = 4, \alpha_2 = 1, \alpha_3 = 1$, while in the data from Fig. 6, we use the value $\alpha_1 = \alpha_2 = \alpha_3 = 4$. The inappropriate metrics doesn't fully consider the role of the road status and size of the cargo, as it emphasizes only the cargo type.

3.3 Computational Efficiency Considerations

In this section, we will briefly address the differences between the naive regular grid approach and adaptive clustering in terms of computational efficiency. The results presented were collected in a single experiment, but are fairly consistent with the other experiments in the series. They are also fairly consistent with the real business environment, where the actors tend to acquire similar services repetitively from the same provider.

While using the general trust model, we only use one trust evaluation per partner – this corresponds to a single reference context. While using the regular grid with an acceptable density over the $\mathbb{C}$, we need a $13^3 = 2197$ ref-

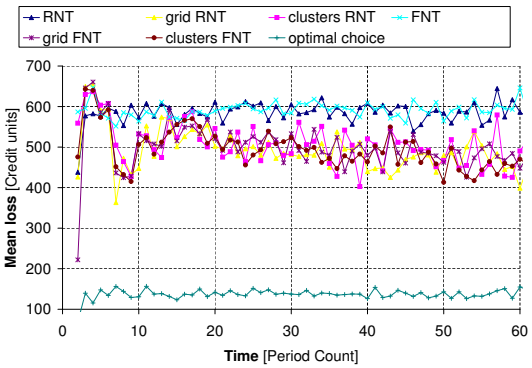

Figure 8. Scenario with trustfulness dependent on all 3 parameters, performance with an incorrect, type-emphasizing metrics. Compare with Fig. 6.

erence contexts (each with a trust model) to model each provider, over a thousand fold increase. While such complex model can be acceptable in some environments, the requirements of the model with reference contexts placed by adaptive clustering are significantly lower – we only need about 12 reference contexts to cover the data and even obtain slightly better results. The number of reference contexts $|\mathcal{R}|$ doesn't only determine the amount of memory, but also the computational efficiency both in the observation and query time. This is due to the fact that the relations 5 and 7 pass through all the points of the $\mathcal{R}$.

4 From Situational Trust to Adaptive Policies

In the above-presented experiments, we have shown that the context modeling is a very viable extension of the trust model that significantly improves the quality of the trusting decisions in the environments where the trustfulness of the agents depends on the situation.

Besides this intuitive interpretation, the model data can be exploited in a more sophisticated manner as well. In the paragraph 3.3, we have mentioned that each trusting decision must integrate the data from the whole set $\mathcal{R}$, introducing a significant performance overhead. To avoid this processing, we may use the model in an "inductive" manner and identify the boundaries of the regions where the partner agents are completely distrustful or completely trustful. Besides the efficiency improvement (that is even disputable when the cutoff distance is selected appropriately), the data regarding the performance of different agents in the similar situations can be analyzed together. When all the agents fail in a certain situation, they may agree to introduce a policy [5] that specifically prohibits such actions, allowing the agents to learn from the experience of their peers without explicitly disclosing their past failures. The implementation of the policy-creating mechanism can be realized either in a peer-to-peer manner, or the policy can be introduced (and possibly enforced) by a dedicated entity (e.g. KAoS server with reputation module) that receives the data from the agents or observes their behavior independently. The example of the rule from our domain is: "Never ship a large quantity of medical supplies at once.".

When considering the rules that doesn't apply to all agents in the community, we can note that the agent identity can be considered as a new dimension of the extended context space $\mathbb{C}$. The main problem of such an extension is the definition of the marginal metrics in this dimension, where we can exploit the techniques from the social network analysis [1, 2] or measure the similarity between agents, while respecting the properties of the metrics[4]. To facilitate the process, we may decompose the single identity dimension into an *identity subspace*, where each agent is defined by one or more crucial properties. The number of considered agent properties determines the subspace dimension.

With this modification, the trust model no longer considers agents as individuals. Rather, following [3], it voluntarily makes the predictions about the performance of the agents by exploiting the data regarding the similar agent's performance in the past. The main advantage is that the extended model learns faster and once the new agent is categorized, its performance can be predicted. This is also a clear advantage in the ad-hoc environments, where there is no agent platform to enforce unique identity of an agent. In the secure environments, the extended model can run in parallel with the classical one and can be used to implement the social/group dimension of trust as defined by [14]. The examples of the rule can be "Don't use agent A for the transport of expensive cargo." in the single-dimension identity case or "Don't use the agents from region North specialized in short-distance hauls for the food transports."

Note that the policies introduced by the mechanism are not necessarily prohibitive; instead, using the principle of the adjustable autonomy [4], the agents may be obliged to obtain an explicit permission before making the positive trusting decision. This corresponds well with the notion of the trust in a highly constrained situation without viable alternatives (i.e. despair [6]).

To infer the rules from the data, we can use a whole range of proven techniques ranging from pattern matching [8], fuzzy control [7] or even formal logic [11].

The ability to define the policies from the data facilitates the integration of trust models with general security infrastructure. Many existing devices or applications can't be retrofitted with trust models of their own and the policies, inferred and communicated by enabled agents, can be

[4] In the trivial case, we can introduce a simple discrete metrics of the type $d(c_i, c_i) = 0\ \forall c_i$ and $d(c_i, c_j) = const\ \forall c_i \neq c_j$.

crucial for the security of the whole system.

5 Conclusions

In our contribution, we have presented an efficient universal method for context representation that can be associated with most currently existing trust models, effectively extending their application from general to situational trust.

An interesting feature of the model is its domain independence – to use the model, user only has to identify relevant features of the situations, define the context space with a corresponding dimension and provide marginal/complete metrics that describes the similarity between various property values. All the subsequent processing is domain and trust model independent, as we also show in the experimental section where we evaluate two different trust models, based on real and fuzzy numbers respectively.

In the experimental part, we shall note that the mechanism turns the trust model into a more machine-learning like problem. We can observe many effects that are typical for learning and classification problems. A good example is a slower learning in higher dimensional problems, where it doesn't depend on the dimension of the space, but rather on the dimension of the subspace containing the data and queries, as shown in Fig. 5 and 7.

In general, situational trust models are more appropriate for the detection of competence component of the trust [6], as they can efficiently distinguish between relevant competencies. On the other hand, we argue that they also provide an inherent resistance against trust building/exploiting, as the high trustfulness acquired in a small contracts is less relevant for high-risk delegations.

The fact that the model provides richer information allows the agents to use it in a more general manner, to autonomously define policies that are adapted to the current state of the environment and can help the agents to avoid bad collaborators without explicitly using the trust model. This extends the applicability of the trust modelling and approaches the trust models with general security infrastructure, which is based on policies and authorizations rather than direct trust use. The trust-enabled agents can be introduced into an existing system to observe its behavior and to share their findings with the other parts of the system by communicating (and/or enforcing) the policies.

The policies can be either identity dependent, or applicable to all agents. If we consider representing the agents by their properties rather than identity, we can obtain a model with inductive properties, able to estimate the performance of new entrants using the experience with the similar partners in the past. The topics outlined in Section 4 will provide us with a vast area for our future research, as we intend to validate the approach on real problems.

Acknowledgment

We gratefully acknowledge the support of the presented research by Army Research Laboratory project N62558-05-C-0028.

References

[1] L. A. Adamic and E. Adar. How to search a social network. *Social Networks*, 27(3):187–203, July 2005.

[2] R. Albert and A. L. Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74:47–97, 2002.

[3] A. Birk. Boosting cooperation by evolving trust. *Applied Artificial Intelligence*, 14(8):769–784, 2000.

[4] J. M. Bradshaw, M. Sierhuis, A. Acquisti, P. Feltovich, R. Hoffman, R. Jeffers, D. Prescott, N. SURI, A. USZOK, and R. Van Hoof. *Agent Autonomy*, volume 7 of *Multiagent Systems, Artificial Societies, and Simulated Organizations*, chapter Adjustable Autonomy and Human-Agent Teamwork in Practice: An Interim Report on Space Applications, page 296. Springer, 2004.

[5] J. M. Bradshaw, A. Uszok, R. Jeffers, and N. Suri. Representation and reasoning for daml-based policy and domain services in kaos and nomads. In *Autonomous Agents and Multi-Agent Systems (AAMAS 2003)*, Melbourne, Australia, 2003. New York, NY: ACM Press.

[6] C. Castelfranchi and R. Falcone. Principles of trust for mas: Cognitive anatomy, social importance, and quantification. In *Proceedings of the 3rd International Conference on Multi Agent Systems*, page 72. IEEE Computer Society, 1998.

[7] D. Driankov, H. Hellendoorn, and M. Reinfrank. *An Introduction to Fuzzy Control*. Springer-Verlag, New York, 1993.

[8] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, New York, NY, USA, 2nd edition, 2001.

[9] D. Huynh, N. R. Jennings, and N. R. Shadbolt. Developing an integrated trust and reputation model for open multi-agent systems. In *Proc. 7th Int Workshop on Trust in Agent Societies*, pages 65–74, 2004.

[10] S. Marsh. Formalising trust as a computational concept, 1994.

[11] M. Pěchouček, J. Tožička, and V. Mařík. Meta-reasoning methods for agent's intention modelling. In *Autonomous Intelligent Systems: Agents and Data Mining*, pages 134–148. Berlin: Springer, 2005.

[12] S. Ramchurn, N. Jennings, C. Sierra, and L. Godo. Devising a trust model for multi-agent interactions using confidence and reputation. *Applied Artificial Intelligence*, 18(9-10):833 – 852, 2004.

[13] M. Rehák, Lukáš Foltýn, M. Pěchouček, and P. Benda. Trust model for open ubiquitous agent systems. In *Intelligent Agent Technology, 2005 IEEE/WIC/ACM International Conference*, number PR2416 in IEEE, 2005.

[14] J. Sabater and C. Sierra. Regret: reputation in gregarious societies. In *AGENTS '01: Proceedings of the fifth international conference on Autonomous agents*, pages 194–195. ACM Press, 2001.

[15] J. Sabater and C. Sierra. Review on computational trust and reputation models. *Artif. Intell. Rev.*, 24(1):33–60, 2005.