The Knowledge Engineering Review, Vol. 00:0, 1–24. © 2016, Cambridge University Press
DOI: 10.1017/S000000000000000 Printed in the United Kingdom

Reputation: A review and unifying abstraction

PHILLIP TAYLOR[1], LINA BARAKAT[2], SIMON MILES[2], NATHAN GRIFFITHS[1]

[1]*Department of Computer Science, University of Warwick, Coventry, CV4 7AL, UK*
E-mail: phillip.taylor, nathan.griffiths@warwick.ac.uk
[2]*Department of Informatics, King's College London, Strand, London WC2R 2LS, UK*
E-mail: lina.barakat, simon.miles@kcl.ac.uk

Abstract

Trust and reputation allow agents to make informed decisions about potential interactions. Trust in an agent is derived from direct experience with that agent, while reputation is determined by the experiences reported by other witness agents with potentially differing viewpoints. These experiences are typically aggregated in a trust and reputation model, of which there are several types that focus on different aspects. Such aspects include handling subjective perspectives of witnesses, dishonesty, or assessing the reputation of new agents. In this paper we distil reputation systems into their fundamental aspects, discussing first how trust and reputation information is represented and second how it is disseminated among agents. Based on these discussions, a unifying abstraction is presented for trust and reputation systems, along with several instantiations to reflect the breadth of models found in the literature. Finally, the abstraction is instantiated to combine the range of capabilities of existing reputation systems in the form of the Machine Learning Reputation System (MLRS), which is evaluated using a marketplace simulation.

1 Introduction

Multi-agent systems (MAS) comprise intelligent agents that interact with each other [42]. Trust and reputation are tools that enable agents to decide whether or not to interact with other agents and to select interaction partners [10, 43]. Trust is the subjective belief, from the perspective of a trustor agent, that a trustee agent will act as they say they will do in a given context [2, 16]. A trustor agent that has a high level of trust in a trustee is confident that an interaction will be successful with good outcomes. Likewise, a low level of trust in a trustee implies that the trustor agent expects an unsuccessful interaction. Whereas trust is assessed using information collected by the trustor, the reputation of a trustee is based on the opinions of several agents in a network.

Trust and reputation systems generally consist of the collection, dissemination, and aggregation of agent experiences [12]. As agents interact with one another they record details of the interaction along with a rating. In centralised systems these records are reported to a central authority which aggregates them and provides reputation assessments of trustees to trustors [16]. In decentralised systems, agents report their experiences and opinions to each other and aggregate them along with their own [38]. In both centralised and decentralised systems several factors affecting reputation assessments must be considered, including lack of experience with a trustee, dishonesty, subjective preferences, and context [10, 12, 43].

In this paper we distil trust and reputation into two kinds of aggregation, namely aggregating interaction records to form trust opinions, and aggregating opinions to produce overall reputation assessments. To this end, the way trust and reputation information is represented is discussed in detail, followed by how it is disseminated while considering dishonesty and preferences. In particular, this paper makes the following contributions:

1. we present a unifying abstraction for the different kinds of trust and reputation system found in the literature,

2. we show for each abstraction how a function of the same form and capabilities can be learned using machine learning over gathered experiences, and
3. we present a machine learning framework to learn trust and reputation models that exhibit the capabilities of existing systems in the literature.

Finally, we propose the Machine Learning Reputation System (MLRS) to combine the capabilities of the different abstractions. The MLRS is then evaluated in a simulated marketplace, against trust and reputation systems that are analogous to those found in the literature.

The paper is structured as follows. First, in Section 2 we review the literature on trust and reputation systems with a focus on how trust and reputation information is represented and disseminated. A unifying abstraction for trust and reputation systems is presented in Section 3, followed by a discussion of its limitations and the corresponding limitations of existing trust and reputation systems in Section 4. Next, Section 5 presents a machine learning framework for building trust and reputation models and introduces the MLRS. Finally, an evaluation of several strategies in a simulated martketplace is presented in Section 6, and Section 8 concludes the paper.

2 Related work

Trust and reputation systems can be divided in many ways, such as whether they are centralised or decentralised, whether they process continuous or discrete ratings, or if they are suitable for dynamic environments or not. Hoelz and Ralha [12] provide a meta-model for trust and reputation systems consisting of three components, namely trust, reputation, and exploration. The trust and reputation components consist of methods for gathering and processing direct and indirect experience, whereas the exploration component contains methods for exploring the environment to locate new trustee agents and reputation information sources. In this paper we consider only the trust and reputation components, and in particular we consider the *representation* and *dissemination* of trust and reputation information. When presenting abstractions for trust and reputation assessment models in Section 3, trust information is generalised to *opinions*, and dissemination is further divided into *gathering* and *aggregating* opinions.

2.1 Representation of trust information

A trust or reputation score is typically given to the user as a single number, representing the trustworthiness of a trustee agent. To compute a trust or reputation score, ratings of past interactions are typically aggregated to produce a set of parameters that form an opinion. In the most simple cases the aggregation computes the mean rating, which is then used as the measure of trustworthiness [13, 14, 30, 31]. Other aggregations produce opinions with the parameters of probability density functions (PDF), such as the Gaussian [38], Beta [15], or Dirichlet PDFs [29, 38]. In these cases, the measure of trustworthiness is typically computed as the expected value of the PDF.

The representation of trustworthiness depends on the domain, as does the kind of ratings that are assigned to interactions. In FIRE [13, 14], for example, ratings are assigned as real numbers in the range $[-1,1]$. This means that the most suitable opinion representation is one with continuous parameters. In FIRE, the mean rating is used as the opinion, which is in fact equivalent to a Gaussian representation where the standard deviation is ignored. In BRS [15], ratings are given as binary (i.e. $\{0,1\}$) to represent a successful or unsuccessful interaction. In this reputation system, therefore, a Beta PDF is most appropriate with α being the number of successful interactions and β the number of unsuccessful interactions. Regan et al. [29] extend this binary rating system to include more values, such as $\{Very\ bad, Poor, OK, Good, Very\ good\}$, and use a Dirichlet PDF as the opinion.

In a stable environment, where the behaviour of agents does not change over time, simple aggregations of historical interaction ratings can be used in assessing trust and reputation. If there are dynamic properties to the environment and its members, however, some historical interactions may be more relevant than others. If the performance of a trustee drifts over time, for example, the ratings given by trustors for its interactions may decrease. Here, an old interaction rating is less relevant to the performance of the trustee after the drift, than a more recent one. In several reputation systems, including REGRET [30],

FIRE [13, 14], and BRS [15], this recency is considered by weighting the interaction records in the aggregation, giving more weight to more recent interactions. In other systems, such as TRAVOS [37] and HABIT [38], interaction ratings older than a threshold are ignored.

Weighting or ignoring interaction ratings by their recency is appropriate when older interactions are always less relevant than more recent ones, but this is not always the case. If there is context associated with interactions, it may be more useful to use this to determine their relevance to the assessment. This notion of contextual trust has been investigated in even the first computational trust models. Marsh [24], for example, considered general trust of an agent separately from situational trust of a trustee in particular situation. The POYRAZ [33] reputation system uses an ontology to describe and compare contexts, and uses only those ratings from relevant interactions in the assessment. If an old interaction occurred under the same context as the one being evaluated, it is highly relevant to the assessment. If the interaction occurred under a completely different context, no matter how recent, is unlikely to be relevant in the trust or reputation assessment. Similarly, Urbano et al. [39] compare the current context to the context extracted from negative experiences with the trustee. If there is a match above a threshold, the trust score is zero and otherwise another trust aggregation is used over all interaction records involving the trustee. Both these context aware assessments may be extended to provide a measure of context similarity, which can replace the recency weighting.

An alternative to simple aggregation is to apply machine learning over features extracted from interaction records, to produce models that predict the ratings [23]. In general, a machine learning algorithm operates over a set of training samples, each with input values and an output target value. The algorithm then learns a model that can accept inputs of the same form, and produce estimates of the target. In the context of trust and reputation, the input values are a vector of features describing the interaction (i.e. features describing the trustee and interaction context, etc.), and the output target value is the rating. Once a learning algorithm has been applied to training samples extracted from interaction records, the learned model can then be used to make predictions about potential future interactions. The most common application of machine learning for trust and reputation is in deriving stereotype trust and reputation [6, 7].

Stereotype trust and reputation relies on agents exhibiting traits prior to an interaction, where those traits are indicative of their behaviour during an interaction. For example, a taxi service may be an airport transfer service or a local taxi, which exhibit different traits that are observable prior to using their services. The airport transfer taxi may have a large storage space, whereas the local taxi may be able to carry more passengers. Other traits include those extracted from the organisational structure in which a trustee is situated or their relationships with other trustees [3]. If several trustee agents with similar behaviours exhibit similar traits, a stereotype can be formed to bootstrap or enhance the reputation assessment process.

Stereotype information is useful in highly dynamic environments where agents join and leave with high frequencies, meaning that trustor agents may have too few direct experiences with trustees to make reliable trust assessments. As with interaction context, one method for considering stereotype information is to use ratings from historical interaction records where similar traits were observed. This is akin to the clustering approach that Liu et al. [19, 21, 22] proposed in StereoTrust, where interactions are separated into groups defined by observed traits. The interactions in groups that match the current trustee traits are then aggregated to produce an overall trust value.

In using stereotype information alongside BRS, Burnett et al. [6, 7] propose a model that learns a mapping from stereotype traits to a trust value. The output of the stereotype model is then used to shift the Beta PDF, also considering how many direct experiences the trustor has had with the trustee. If the trustor has directly interacted with the trustee only a small number of times, the distribution is shifted to have a mean close to the output of the stereotype model. When the trustor has an abundance of experience with the trustee, the stereotype model output has little impact on the overall assessment in comparison to the direct experience.

When a trustor has not observed a trustee directly nor any of its traits, it has no information with which to make a trust assessment. To overcome this lack of information it is typical to request reputation information from witnesses, which is discussed in detail in Section 2.2. An alternative to this is to

generalise stereotype traits so that a stereotype model can be built with even small numbers of interaction records. Fang et al. [8] present a fuzzy decision tree that learns from a semantic ontology of stereotypes that is able to generalise traits when making assessments of stereotype trust. For instance, if the trustor has experience with trustees that have the trait of their location being China, this can be generalised to Asia in the ontology. With this generalisation, the decision tree is then able to make an assessment of stereotype-trust for a trustee in Japan even when there is no experience of trustees in this location.

Some representations of trust and reputation, in particular those that are based on PDFs, also provide an accessible measure of confidence for the trust score. Teacy et al. [37] define confidence in a Beta PDF as the integral of a range around the expected value, which increases as more evidence is gathered and $\alpha + \beta$ increases. This confidence measure can be used to evaluate how reliable the trust score is, in particular when deciding whether or not to use other sources of information (see Section 2.2). Rather than confidence, Jøsang and Ismail [15] propose a measure of *uncertainty* for the Beta PDF, which decreases as more evidence is gathered. Other measures of confidence may be based on the variance of the distribution, particularly if the representation used is a Gaussian PDF. In general, a PDF that has a high variance is considered to be less reliable than a PDF with low variance, which should be reflected in the confidence measure.

For opinions produced by machine learning, Liu et al. [23] discuss using the estimated performance of the model along with characteristics of the input to provide a measure of confidence in the trustworthiness score. In particular, confidence in assessments should be low for models that are known to perform poorly in evaluation. A learned model can be evaluated using either *k*-folds cross validation, or by using testing samples that are distinct from the training samples [41]. Furthermore, several machine learned models, including decision trees and naïve Bayes, provide confidences alongside their predictions.

Finally, trustworthiness is sometimes considered to have multiple dimensions, such as when interactions are rated using multiple terms. In a logistics scenario, for example, there may be two term dimensions, such as timeliness of delivery and politeness. If a trustor does not mind whether the trustee is polite or not, then it can look at timeliness only, but if both timeliness and politeness matter then both should be considered. In this paper, we make the simplifying assumption that ratings, trust and reputation have only one dimension. Furthermore, a multidimensional rating can be converted to have a single dimension by taking a weighted average, or multiple reputation models can be created for each dimension and their outputs averaged [9].

2.2 Dissemination of reputation information

In most trust and reputation systems the main source of information is the direct experience of the trustor agent. When a trustor has too little direct experience, they can gather information from other sources. Typically, these other sources are witness agents, or advisors, who report on their experiences to help the trustor assess reputation. Another source of reputation information is from reference agents selected by the trustee [13, 14]. There are two main considerations when disseminating reputation information, first is how reputation is communicated from witnesses to the trustor, and second is how to handle dishonesty of witnesses, their personal preferences, and different perspectives.

Reputation systems generally fit into one of two broad categories, ether centralised or distributed. In centralised systems, reputation information, typically in the form of raw ratings, is transmitted to a central authority for processing. In the SPORAS reputation system [44], trustor agents provide ratings to a central authority, which then updates the reputation score for the trustee accordingly. Once processed, trustor agents can request this reputation information to inform their decisions of whether or not to interact with a trustee. The processing can be simply collating witness ratings for the trustor to aggregate, or it can be an aggregation in itself with the result provided to the trustor.

Whereas centralised systems require a central authority to be maintained, reputation information in decentralised systems is communicated directly between agents in the network. A trustor typically requests information from witnesses, who then report their opinions to the trustor for combination with their own direct evidence. To make requests the trustor must first discover witnesses, which can be found either via a public directory, if one exists, or by asking agents the trustor has knowledge of (such as those

they have previously interacted with). This is the basis of TrustNet [32], in which a request for reputation information is sent to agents the trustor has previously interacted with. These witnesses are then able to make requests on behalf of the trustor within their neighbourhood, creating a network of trust to be combined into a single reputation assessment.

The FIRE [13, 14] reputation system has a similar reputation dissemination mechanism, where agents within a local radius on a map are to be considered neighbours. Requests are then made to neighbours of the trustor, who pass them onto their neighbours recursively until sufficient reputation information is gathered to make an assessment. Likewise, in DTMAS [1] a breadth first search of the agent network is performed, up to a given depth decided by the trustor.

When reputation information is gathered, it must then be combined along with any direct information the trustor has. If witnesses are known to be honest and to rate interactions in the same way as the trustor, the opinions can be combined using either a sum of functions or sum of parameters approach [20]. In a sum of functions, the expected value of each of the gathered opinions is computed and the results combined to produce an overall reputation score. When opinions all have the same representation (such as Beta or Dirichlet PDFs) a sum of parameters can be used, to produce an overall opinion.

Reports provided by witnesses that are dishonest or otherwise rate interactions differently to the trustor can be detrimental to reputation assessment if they are trusted blindly. The FIRE reputation system is able to control the impact that witness reputation has on the overall reputation assessment, but this can lead to reliable opinions being ignored unfairly. Building on BRS and using Beta PDFs, Whitby et al. [40] iteratively remove witness opinions that have expected values that are outliers when compared to all the opinions combined. Teacy et al. [37] developed TRAVOS to discount individual opinions based on the perceived accuracy of the witness. If a witness has previously provided reports that differed significantly from the subsequent interaction ratings, they are deemed to be unreliable. Reports provided by unreliable witnesses are given a low weight so that they affect the overall reputation assessment less than those from reliable witnesses. In STAGE, Şensoy et al. [34] filter unreliable opinions, while using stereotypes extracted from witnesses to determine their reliability.

Discounting reports from witnesses who have previously provided poor information means that they do not negatively affect the reputation assessment. In some cases, where agents rate interactions differently and in a consistent way, however, their ratings may be useful. Reports provided by a witness that always rates interactions inversely from the trustor are useful in reputation assessment once this difference in ratings is understood. Koster et al. [17] provide a argumentation framework for agents to discuss and translate their opinions from one perspective to another. Using Bayesian networks, BLADE [29] and HABIT [38] both aim to re-interpret ratings from such witnesses, and maximize the useful information that is input into reputation computation. BLADE and HABIT both learn the rating function used by witnesses from their ratings of several trustees, comparing them to those of the trustor. Similarly, Koster et al. [18] use shared experiences to learn an alignment for trust models in different perspectives.

3 Model abstractions

Reputation models aim in general to compute the trustworthiness of an agent using information gathered from the environment. The reputation score, $\mathscr{R}_{tr}^{te}$, is often an estimate of the utility that will be gained if an interaction took place between the trustor agent, tr (who is performing the reputation assessment) and the trustee agent, te (whose reputation is being assessed) [14]. In other cases the reputation is the likelihood that the interaction will be successful, or have an outcome above some threshold [7]. The information gathered is often collected as descriptions of interactions that have taken place, involving a trustee, te, and a trustor, tr, or some other witness agent. These descriptions are in the form of tuples, describing the agents that interacted, details of the interaction, and any ratings assigned by tr to the interaction. The ratings represent how good the trustor agent thought the interaction was, and usually are either the utility gained during the interaction or whether or not it was successful.

In this section we discuss abstractions of the kinds of reputation system proposed in the literature. First we discuss the sources of information used in reputation assessments, namely the experience of the trustor and that of witness agents. Second, given these sources of information, we detail a set of trust

Symbol	Description
tr	Trustor agent
te	Trustee agent
w	Witness agent
agt	An arbitrary agent (which can be either a trustor, witness, or trustee in a given interaction)
$\mathbf{W}$	Set of witness agents
$\mathbb{I}_{agt}$	Set of possible interaction experience databases from the perspective of agt
$\mathbb{S}_{agt}$	Set of possible situations from the perspective of agt
$\mathbb{O}_{agt}$	Set of possible opinions from the perspective of agt
$\mathbf{I}_{agt}$	Interaction database expressed from the perspective of agt
I	Interaction tuple
S	Situation tuple
O_{agt}	Opinion expressed from the perspective of agt
Θ	Set of possible trustee traits
θ	A particular set of trustee traits
$\mathbb{C}$	Set of possible user-contexts
C	A particular user-context
$\mathbb{M}$	Set of possible service-contexts
M	A particular service-context

Table 1 Notation used for reputation system abstractions.

functions that aggregate interaction records to provide an opinion, and finally we detail mechanisms for combining opinions provided by different agents. The abstractions we describe are to be combined in different permutations, to produce reputation systems that are robust to different kinds of environment with different requirements. For example, in highly dynamic environments where agents exhibit traits indicative of their behaviour during an interaction, stereotype information may be combined with direct and witness information for more accurate reputation assessments [7, 34].

In general, reputation is computed as,

$$\mathscr{R}(S) = E[u(S)|\mathscr{D}] \tag{1}$$

where $u(S)$ is the utility of the situation (an estimate of prospective interaction ratings), S, being evaluated, and $\mathscr{D}$ is the set of information sources used in the evaluation. A situation is a set of parameters describing a prospective interaction to be assessed by a reputation system. For example, a situation may describe an interaction with a trustee, te, under user-context, C. The reputation system then analyses this against the data it has gathered from the environment, which is typically a set of interaction records, $\mathbf{I}$. Throughout this section, and subsequently in this paper, we will use the notation defined in Table 1.

3.1 Gathering opinions

In this section we discuss the sources of information that reputation assessment is based on. In particular, direct information, which is derived from the experiences of the trustor agent, and witness information, which is provided by agents other than the trustor. In the literature, witnesses are often considered as acquaintances of the trustor, and are typically other trustor agents not involved in the current reputation assessment. Here, we consider a witness as any agent able to provide information that is relevant to the reputation assessment. This may be a central authority, such as a regulatory body, or other competing providers, such as other trustees.

3.1.1 Direct information

Almost all reputation systems use some form of direct information, collected through the trustor agent's own experience. In general, the direct-trust of a trustee in a particular situation is,

$$\mathscr{R}(S) = E[u(S)|\mathbf{I}_{tr}], \tag{2}$$

which estimates the utility of the prospective situation, given the interaction history of the trustor, $\mathbf{I}_{tr}$. Our abstraction for this reputation computation is a function that maps a situation and a set of interaction records to an opinion representation,

$$f_{tr} : \mathbb{S}_{tr} \times \mathbb{I}_{tr} \rightarrow \mathbb{O}_{tr}. \tag{3}$$

For example, a situation may be described simply as a potential interaction between the trustor and the trustee (i.e. $S = \langle tr,te \rangle$), with interaction records consisting of this information along with a rating (i.e. $I \in \mathbf{I}_{tr} = \langle tr,te,r \rangle$). The direct-trust function then processes the interaction records with respect to the situation to output an opinion, O_{tr}.

In general, in this paper, we consider a situation tuple, $S = \langle tr,te,\tau,\theta,C \rangle$, to contain all information known during a reputation assessment prior to an interaction. This includes the interacting parties, tr and te, the current timestamp, τ, the trustee stereotype traits observed by the trustor, θ, and the known-context, C. Interaction records, $I = \langle tr,te,\tau,\theta,C,M,r \rangle$, contain the situation definition along with unforeseen-context information, M, and the rating given by the trustor, r, which are only known after the interaction has taken place. For both situation and interaction tuples an element is indexed using a subscript, for example the trustee agent in an interaction, I, is I_{tr}.

In a typical reputation system, such as FIRE or BRS, this function searches through interaction records to find those that match the current situation (i.e. those with the same te). These records are then aggregated to produce an opinion, which in FIRE is a real number in the range $[-1,1]$, and in BRS is a Beta PDF represented by α and β parameters. In the general case, a second function,

$$f_{tr}^{rep} : \mathbb{O}_{tr} \rightarrow \mathbb{R}, \tag{4}$$

is applied to convert the opinion to a real valued trust score. In FIRE, this function may simply output the opinion, whereas in BRS this function queries the Beta PDF for its expected value.

3.1.2 Witness information

In assessing reputation, the trustor may have little or no direct information to make a direct-trust assessment and must acquire information from other sources, such as witnesses. When combined with direct information, the reputation of a trustee in a particular situation is,

$$\mathscr{R}(S) = E[u(S)|\mathbf{I}_{tr},\mathbf{I}_w \forall w \in \mathbf{W}], \tag{5}$$

where $\mathbf{W}$ is the set of all witness agents that provide information for use in the reputation assessment.

We define a witness as any agent that is not the trustor providing information for use in the reputation assessment. Typically, witnesses are synonymous with advisor agents who are acquaintances or neighbours of the trustor, but this can be extended to those gathered through a recursive search of the trustor's social network. In FIRE, another kind of witness is introduced, selected by the trustee as a referee, to provide information for certified reputation. Although the information provided by these referee witnesses cannot be tampered with by the trustee, it is likely that only referees with positive opinions will be selected. This is therefore likely to result in certified reputation being optimistic with respect to the true reputation score, which should be handled appropriately by the aggregation.

As with direct-trust, each witness has their own individual trust function,

$$f_w : \mathbb{S}_w \times \mathbb{I}_w \rightarrow \mathbb{O}_w, \tag{6}$$

which maps situations and interaction records to opinions, all from their own perspective and in their own representations. After applying this function, the witnesses then provide their opinions to the trustor to

combine them along with their direct-trust score. Such a combination function has the form,

$$f^{agg} : \mathbb{O}_{tr} \times \prod_{w \in \mathbf{W}} \mathbb{O}_w \to \mathbb{O}_{tr}, \tag{7}$$

mapping all gathered opinions onto an opinion in the representation used by the trustor. Finally, the opinion is processed in the same way as in Equation 4 to produce a single reputation score, $\mathscr{R}_{tr}(S_{tr})$.

3.2 Opinion representation

This section outlines a representative selection of opinion representations. First, a simple aggregation of ratings is considered, which is then built upon to incorporate more complex facets of reputation assessment, including stereotypes and contexts. Each element of the situation and interaction tuples may be irrelevant for a particular environment or opinion representation, and may be omitted when appropriate.

3.2.1 Basic trust

A simple method, used in FIRE, for computing an opinion is to take a weighted mean over all ratings from interactions that match the prospective situation,

$$O_{tr} = f_{tr}(S, \mathbf{I}) = \frac{\sum_{I \in \mathbf{I}} \Psi(S,I) \cdot I_r}{\sum_{I \in \mathbf{I}} \Psi(S,I)}, \tag{8}$$

where the interaction weight,

$$\Psi(S,I) = \begin{cases} 1 & \text{if } S_{te} = I_{te} \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

As this aggregation represents opinions as a single real number, it is appropriate to also use this as the trust score, i.e. $\mathscr{R}_{tr}(S_{tr}) = O_{tr}$.

When ratings are binary, as in BRS, the parameters of a Beta PDF can be used to form the opinion,

$$O_{tr} = f_{tr}(S, \mathbf{I}) = \langle \alpha, \beta \rangle, \tag{10}$$

where,

$$\alpha = \sum_{I \in \mathbf{I}} \begin{cases} \Psi(S,I) & \text{if } I_r = 1 \\ 0 & \text{otherwise} \end{cases}, \text{ and } \quad \beta = \sum_{I \in \mathbf{I}} \begin{cases} \Psi(S,I) & \text{if } I_r = 0 \\ 0 & \text{otherwise} \end{cases}, \tag{11}$$

represent the number of matching positive and negative ratings respectively. While there are more concise ways of defining α and β (e.g. using set cardinality), the summations of interaction weights enable BRS to be sensitive to recency, stereotype, and context information, as discussed later in this section. This representation is used by all reputation systems that are based on BRS, including TRAVOS [37] and STAGE [34]. As previously stated, where opinions are represented as a set of parameters, a second trust function is required,

$$f_{tr}^{rep}(O_{tr}) = \frac{\alpha + 1}{\alpha + \beta + 2}, \tag{12}$$

which calculates the expected value of the Beta PDF and transforms the opinion into a trust or reputation score.

Opinions can be represented using different sets of parameters, including those for Dirichlet PDFs [29, 38] or Gaussian distributions [38]. Due to space constraints we do not provide a comprehensive set of opinion representations, but the abstraction presented is able to accommodate any opinion representation that can be queried for a trust value (such as the expected value of the distribution they represent). As an example, the parameters of a Dirichlet PDF may represent the number of times a particular rating out of a set of discrete values (e.g. $\{Very\ poor, Poor, Good, Excellent\}$) has been given in previous interactions. The expected value and reputation or trust score is then computed as the most common rating. It is also possible to represent trust in multiple dimensions by using multiple sets of parameters in an opinion. For example, both the quality and timeliness of a service may be represented as two Beta PDFs in one opinion, i.e. $O_{tr} = \langle \alpha_{\text{quality}}, \beta_{\text{quality}}, \alpha_{\text{timeliness}}, \beta_{\text{timeliness}} \rangle$. Our abstraction only assumes that the final reputation score is represented as a single value.

3.2.2 Weighting for recency

Many reputation systems consider recent information to be more important than old information. This is typically achieved through applying weights to ratings in the aggregation proportionally to their age, which can be incorporated into the abstractions interaction weighting function,

$$\Psi_{rec}(S,I) = \begin{cases} \lambda_{rec}(S,I) & \text{if } S_{te} = I_{te} \\ 0 & \text{otherwise,} \end{cases} \tag{13}$$

where $\lambda_{rec}(S,I)$ is a recency function that outputs a larger value for more recent interactions. The FIRE reputation system uses a recency function that gives logarithmic weights based on the age of the interaction,

$$\lambda_{rec}(S,I) = e^{-\frac{\Delta(S_\tau, I_\tau)}{\lambda_d}}, \tag{14}$$

where $\Delta(S_\tau, I_\tau)$ is the time since interaction I, and λ_d is a constant defining the amount of decay [14]. Of course, this weighing mechanism can also be used when aggregating the α and β parameters of BRS.

In some reputation systems old information is simply ignored, giving a interaction weighting of,

$$\Psi_{rec}(S,I) = \begin{cases} 1 & \text{if } (S_{te} = I_{te}) \wedge (\Delta(S_\tau, I_\tau) \geq \lambda_\tau) \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

Here, λ_τ is a cut-off for ignoring interaction ratings, where interactions older than the cutoff are given a weight of 0 and not considered in forming the opinion.

3.2.3 Stereotypes

In many domains, agents exhibit traits that are indicative of their behaviour during an interaction, and are observable during reputation assessment. If several trustee agents with similar behaviours exhibit similar traits, a stereotype can be formed to bootstrap or enhance the reputation assessment process. In general there are two techniques to incorporate stereotype information into the trust functions, namely to adapt the match function to output stereotype similarity, or to use machine learning in the trust function.

The interaction weight when taking stereotype information into account should not depend on whether the trustees are the same or not, but rather on the similarity of their stereotype features. For example, the Jaccard index may be used to measure the similarity of the two sets of trait observations.,

$$\Psi_{st}(S,I) = \frac{S_\theta \cap I_\theta}{S_\theta \cup I_\theta}. \tag{16}$$

As with recency, a threshold may also be used alongside a similarity measure such as the Jaccard index, where the interaction weight is 1 when the similarity is above the threshold and 0 otherwise. This thresholding method is akin to the clustering approach that Liu et al. [22] proposed in StereoTrust, where interactions are separated into groups defined by observed traits. The interactions in groups that match the current trustee traits are then aggregated to produce an overall trust value.

An alternative technique is to apply machine learning to build a predictive model, taking observed traits as the inputs and outputting an estimate of trust. A stereotype function, $f^{st}(\cdot)$, is built using learning algorithms such as naïve Bayes or the MD5 decision tree [28]. A training set can be generated from interaction records, with the stereotype traits as inputs and the ratings as outputs. In particular, Burnett et al. [6, 7] use the MD5 learning algorithm and generate one training sample per trustee, with the traits as inputs, and the output variable is the direct trust computed using Equation 12. This gives a small training set to learn from efficiently, but assumes that trustee traits do not change over time. An alternative to this is to generate a sample for each interaction, but this may bias the trust value of a stereotype to a popular trustee with numerous interactions. Once learned, observed traits, S_θ, are used as the input to estimate the stereotype-trust, $f^{st}(S_\theta)$, of a trustee.

This output can then be combined with more simple aggregations, as in Burnett et al. [6, 7], Şensoy et al. [34], and Taylor et al. [35, 36], or the model used directly as the opinion. Burnett et al. [6, 7] extend

BRS to include stereotypes, and produce an opinion,

$$O_{tr} = f_{tr}(S, \mathbf{I}) = \langle \alpha, \beta, f_{tr}^{st}(S_\theta) \rangle, \tag{17}$$

where the stereotype-trust, $f^{st}(S_\theta)$, is the likelihood of a successful interaction after observing the trustee traits, S_θ. An overall trust value is then computed as,

$$f_{tr}^{rep}(O_{tr}) = \frac{\alpha + 2 f_{tr}^{st}(S_\theta)}{\alpha + \beta + 2}, \tag{18}$$

When $\alpha + \beta \approxeq 0$, this prior provides the trust function with a default value for trust, based on the observed traits of the trustee. To combine this opinion with witness opinions, the mean value of each of the parameters can be taken to form a new opinion. With no stereotype information, this equation becomes equivalent to BRS in Equation 12 (i.e. when $f_{tr}^{st}(S_\theta) = 0.5$).

3.2.4 Known context

When the context of a prospective interaction is known and is part of the situation tuple, it should be accounted for in the reputation assessment. One approach is to alter the interaction weighting function so that it reflects the similarity of the interaction record to the current context,

$$\Psi_{ctx}(S, I) = \begin{cases} \lambda_{ctx}(S, I) & \text{if } S_{te} = I_{te} \\ 0 & otherwise, \end{cases} \tag{19}$$

where the Jaccard index is again used to measure context similarity,

$$\lambda_{ctx}(S, I) = \frac{S_C \cap I_C}{S_C \cup I_C}, \tag{20}$$

where S_C and I_C is recorded as a set of traits. As with stereotype and recency weighting, a threshold can be applied, where the interaction weight is 1 when the similarity is above the threshold and 0 otherwise.

Another approach is to have a different opinion for each context [38],

$$O_{tr} = f_{tr}(S, \mathbf{I}) = \langle S_C, O_{tr}^{C_1}, O_{tr}^{C_2}, \ldots O_{tr}^{C_n} \rangle, \tag{21}$$

where $O_{tr}^{C_1}, O_{tr}^{C_2}, \ldots, O_{tr}^{C_n}$ are contextual opinions for contexts, $C_1, C_2, \ldots, C_n$, each produced using aggregations over interaction records with a context sensitive interaction weighting. A trust score can be computed using the most appropriate contextual opinion or as an aggregation of all the opinions, weighted by their context similarities.

As with stereotype trust, a third approach to incorporate user context is to apply machine learning and to use the context fetaures as inputs to a context model. The output of this model can then be either used directly as the opinion or as a parameter in the trust function. This is discussed further in Section 5.

3.2.5 Unforeseen context

When the context is only known after an interaction has taken place, and not during the assessment of a situation, it cannot be considered directly. Instead, the impact of contexts on previous interactions must be considered when inspecting their ratings. We consider two kinds of unforeseen context, namely those that were caused by the trustee and those that were caused by external factors. If the context is caused by the trustee, then the interaction ratings are relevant when performing reputation assessment. For instance, if a logistics company is delayed after taking on a second contract, a poor rating for timeliness may be relevant and should be considered. If an external factor is the root cause of the context, such as an unexpected storm causing the delay, it may be more reasonable to ignore a poor rating. Similarly, a good rating caused by an external factor should also be ignored.

To consider these unforeseen contexts that should be ignored in reputation assessment, Miles and Griffiths [25, 26] adapted the recency weighting of FIRE to reflect the impact of a context on an

interaction. An example interaction weighting function for the abstraction is,

$$\Psi_{mit}(S,I) = \begin{cases} 1 & \text{if } (S_{te} = I_{te}) \wedge (I_M = \text{normal}) \\ 0.1 & \text{if } (S_{te} = I_{te}) \wedge (I_M \neq \text{normal}) \\ 0 & \text{otherwise,} \end{cases} \tag{22}$$

where an interaction unforeseen context, I_M, is normal when the context was under the control of I_{te}, and 0.1 is arbitratily used to represent a low weight.

3.3 Opinion aggregation

To combine direct and witness information the opinions must be aggregated. One simple aggregation, when all opinions contain the same parameters, is to take the mean of each opinion parameter value,

$$f^{agg}(O_{tr}, O_w \forall w \in \mathbf{W}) = \frac{1}{|\mathbf{W}|+1}\left[O_{tr} + \sum_{w \in \mathbf{W}} O_w\right], \tag{23}$$

where the addition of two opinions is the addition of their parameters. For example, the addition of two BRS opinions, $O_a = \langle \alpha_a, \beta_a, \rangle$ and $O_b = \langle \alpha_b, \beta_b, \rangle$, is $O_a + O_b = \langle \alpha_a + \alpha_b, \beta_a + \beta_b, \rangle$. A trustor may wish to give more importance to their own direct information compared to witness information, and so this aggregation may be weighted to reflect this,

$$f^{agg}(O_{tr}, O_w \forall w \in \mathbf{W}) = \frac{1}{|\mathbf{W}|+1}\left[\omega_{dir} O_{tr} + \omega_{wit} \sum_{w \in \mathbf{W}} O_w\right], \tag{24}$$

where ω_{dir} and ω_{wit} are weights for the direct and witness information respectively. Similarly to addition, multiplication of an opinion by a scalar is applied to each of the opinion parameters, i.e. $\omega O_a = \langle \omega\alpha_a, \omega\beta_a \rangle$. This weighted aggregation is similar to that used in FIRE, where the trustor defines the weights to sum to one.

In general it is unreasonable to assume that all witnesses will be benevolent and provide opinions from the same perspective as the trustor. To combat this, many reputation systems, such as TRAVOS [37] and STAGE [34], discount witness opinions to limit their impact on the overall reputation assessment, particularly if they differ subjectively. Other reputation systems, including BLADE [29] and HABIT [38], map opinions from the witness perspective to the perspective of the trustor, accounting for differing opinion representations and preferences.

3.3.1 Discounting witness information

If witnesses are known to be unreliable, it is possible to ignore their information in Equation 24 by setting the witness weight to zero, $\omega_{wit} = 0$. Universally discounting all witnesses ignores some potentially reliable information, however, and so TRAVOS [37] uses an individual weighting for each witness,

$$f^{agg}(O_{tr}, O_w \forall w \in \mathbf{W}) = \frac{1}{|\mathbf{W}|+1}\left[\omega_{tr} O_{tr} + \sum_{w \in \mathbf{W}} \omega_w O_w\right], \tag{25}$$

based on the opinions they have previously provided. The individual weights again sum to one, and are derived by comparing the opinions of trustees provided by the witnesses to what happened during subsequent interactions with the same trustees. Specifically, the witness weights in TRAVOS shift the Beta PDF toward the uniform distribution with $\alpha = \beta = 1$ in proportion with the disagreement between the witness and trustor opinions.

3.3.2 Reinterpreting witness information

When witnesses disagree with the trustor in a consistent way, then discounting or ignoring their opinions may be detrimental to reputation assessment. For example, if a witness is observed to consistently have the opposite opinion to the trustor, this knowledge can be used to reinterpret or flip future opinions provided

by the witness. Similarly, opinions of a witness that are consistently optimistic about the performance of trustees can be mapped to be more realistic. The opinion aggregation function for a reputation system with this reinterpretation capability is,

$$f^{agg}(O_{tr}, O_w \forall w \in \mathbf{W}) = \frac{1}{|\mathbf{W}|+1}\left[\omega_{tr} O_{tr} + \sum_{w \in \mathbf{W}} \omega_w \rho(O_w)\right], \tag{26}$$

where $\rho(\cdot)$ is a mapping function that reinterprets the witness opinion. The weighting, ω_w, remains in this definition for witnesses that are unreliable in an unpredictable way.

The mapping funtion, $\rho(\cdot)$, can be learned by inspecting the opinions provided by a witness against those of the trustor. For example, if the witness has provided opinions for several different situations for which the trustor also has sufficient information from which to form an opinion, a mapping between the two perspectives can be built. Existing reputation systems that have this reinterpretation capability include BLADE [29] and HABIT [38], which are based on Bayesian networks that model the rating functions of witnesses and map their opinions to the trustor's perspective.

4 Limitations

Trust and reputation is a multi-faceted problem, of which most trust and reputation systems target one particular aspect. As a result, most reputation systems in the literature are limited as they do not easily generalise to new environments with new information types or sources. In our abstraction we have provided a mechanisms for including different types of information in aggregation models. The FIRE and BRS models, for example, were extended to use context information by altering the interaction weighting function, as in Equation 19.

While including information using weighting functions works in isolation, it is unclear how multiple types of information should be combined using aggregation models. One approach may be to aggregate the weights themselves, for example using a multiplication, but the distribution of weights may be complex. Multiplying weights that are each in the range $[0,1]$ may also produce very small values for overall weights. This issue could be solved by separating the filtering and weighting of interactions, but this adds unnecessary complexities into the abstraction that may limit its applicability in other ways.

Another limitation of many reputation systems is that they use endogenous discounting, causing a feedback loop where useful witness information is discounted because it disagrees with unreliable direct information [27]. This is not accounted for in our abstraction, as it relates to the computation of opinion weights and formation for reinterpretation functions, rather than their application.

5 Machine learning of trust functions

In this section we view reputation assessment as a machine learning problem, and discuss how a trust function can be learned algorithmically to form opinions. Using machine learning, descriptions of interactions, their context, and the environment, can easily be incorporated in assessing reputation by appending features to the samples. In addition, viewing reputation assessment as a machine learning problem enables the use of established and well understood techniques to enhance the learning process, including feature selection, cross-validation, and ensemble learning [41]. If context features extracted from interaction tuples are not relevant to the ratings then they can be ignored by the machine learning process, especially if feature selection techniques are applied.

In general, a supervised learning algorithm, $F(\cdot)$, builds a function, $f^{ml}(\cdot)$, that maps m input features to a target variable. The algorithm learns this function by processing training data with n samples, that each have m features and a target variable. In the case of reputation assessment, the training data is generated from interaction records, with m features extracted to describe an interaction and a target variable of the rating. The training data for the basic trust function using direct information, comprises samples extracted from all interactions tr has taken part in, $\mathbf{I}_{tr}$. Each interaction record produces one sample with values, $\langle te; r\rangle$, where te is an input feature signifying the trustee in the interaction, and r is the target value representing the interaction rating. A learning algorithm is applied to the training samples, $F(\forall\langle te; r\rangle \in \mathbf{I}_{tr})$, to produce the mapping function, $f^{ml}_{tr}(\cdot)$. The reputation of different trustee agents is then assessed

Trust function	Input structure	Similar Models	Notes
Basic trust	$\langle te; r\rangle$	BRS	Sample for each interaction between the trustor or witness agent and a trustee.
Recency	$\langle te; \Psi_{rec}(S,I) r\rangle$	REGRET, FIRE	Sample for each interaction between the trustor or witness agent and a trustee, with training samples weighted by their age.
Stereotypes	$\langle \theta; r\rangle$	Burnett, STAGE, POSSTR	Sample for each interaction between the trustor or witness agent and a trustee. This is to be combined with other models, either by adding te to the samples or by aggregating the outupts.
Known context	$\langle te, C; r\rangle$	POYRAZ	Sample for each interaction and witness reports, including features describing the known context.
Unforeseen context	$\langle te; \Psi_{mit}(S,I) r\rangle$	Miles	Sample for each interaction and witness reports, with training samples weighted by the unforseen contexts.
Combined trust	$\langle te, \theta, C; \Psi_{mit}(S,I) r\rangle$	N/A	This is a combination of the above models, with training samples weighted by the unforseen contexts.

Table 2 Sample structures extracted from interaction records that can be used in a supervised machine learning algorithm. The input features and target value of the sample structures are separated by ';'.

by inputting to the learned function samples of the form $\langle te; ?\rangle$, extracted from situation tuples describing the potential interactions between tr and the different trustees. The estimated utility for the situation, $E[u(S)|\mathbf{I}_{tr}] = f^{ml}_{tr}(\langle te\rangle \in S)$, is then the trust score for te in the prospective interaction.

Table 2 lists each of the trust functions discussed in Section 3.2, presenting the structures of input data for their corresponding machine learning models. For clarity, input features are each separated by ',', which are separated from the target value by ';'. For recency and unforeseen context the training samples are weighted prior to applying the learning algorithm. In general, an all-or-nothing weighting function can be used to filter out those training samples that are too old (as in Equation 15) or where an unforeseen context occurred. Alternatively, several machine learning algorithms are able to account for training weights, by learning more from training samples with high weights and less from those with low weights.

Hereinafter, we do not repeat the input features when referring to a learning function or the learned models. This is both for clarity and to abstract the input structures of the models, which are independent of the learning algorithm and combining opinions. A opinion learned using direct information, $\mathbf{I}_{tr}$, is therefore,

$$O_{tr} = \langle f^{ml}_{tr}(S)\rangle, \tag{27}$$

which can be defined using a partial function application,

$$f^{ml}_{tr}(S) = F(\mathbf{I}_{tr})(S). \tag{28}$$

Similarly, an opinion learned by a witness, w, is,

$$O_w = \langle f^{ml}_{w}(S)\rangle = \langle F(\mathbf{I}_w)(S)\rangle. \tag{29}$$

5.1 Opinion aggregation

To combine direct and witness information with trust functions of this kind, there are two possibilities. First, the learned functions can be stored in the opinion tuple and then combined in the opinion aggregation, either by collating the interaction records or by combining the model parameters. Second, the outputs of the learned functions comprise the opinions and these can be aggregated to produce an overall reputation score. For simplicity, in this paper we mainly consider aggregating the model outputs, as this is more amenable to discounting and reinterpretation of opinions. Specifically, the overall reputation score

is the mean output of the direct and witness models,

$$f^{agg}(O_{tr}, O_w \forall w \in \mathbf{W}) = \frac{1}{|\mathbf{W}|+1}\left[f_{tr}^{ml}(S) + \sum_{w \in \mathbf{W}} f_w^{ml}(S)\right]. \tag{30}$$

5.2 Discounting witness opinions

To discount unreliable witness opinions we use,

$$f^{agg}(O_{tr}, O_w \forall w \in \mathbf{W}) = \frac{1}{|\mathbf{W}|+1}\left[\omega_{tr} f_{tr}^{ml} + \sum_{w \in \mathbf{W}} \omega_w f_w^{ml}(S)\right], \tag{31}$$

where ω_{tr} and ω_w are weights that reflect the performance of the respective learned models. The performance can be measured in accuracy, true positive or true negative rates, precision, or recall, etc., and can be estimated using a k-folds cross validation or another evaluation procedure [41]. In keeping with the TRAVOS discounting mechanism, the weights in this paper represent the ability of the learned model to predict ratings given by trustor. Specifically, $\omega_{tr} = 1$, and each witness weight is the accuracy of the witness model estimated using the direct interactions where trustor has previously given a rating. For discrete rating systems this is the number of true positives when predicting the rating divided by the number of direct interaction records. With continuous ratings, the accuracy may be derived from the root mean squared error.

5.3 Reinterpreting witness opinions

To reinterpret witness opinions, an individual reinterpretation model, $\rho_w(\cdot)$, is learned for each witness, $w \in \mathbf{W}$, to reinterpret the outputs of the witness models to the perspective of tr. The training samples for a reinterpretation model are generated from both the interaction records and witness reports, $\mathbf{I}_{tr} \cup \mathbf{I}_w$. Each of these records is used to generate a reinterpretation training sample, $\langle f_w^{ml}(I); f_{tr}^{ml}(I)\rangle$, with an input of the witness opinion and a target of the trustor opinion. The reinterpretation model learned,

$$\rho_w(\cdot) = F(\forall \langle f_w^{ml}(I); f_{tr}^{ml}(I)\rangle \in \mathbf{I}_{tr} \cup \mathbf{I}w) \tag{32}$$

is then able to convert an input rating from the perspective of the witness, O_w, to the perspective of the trustor, O_{tr}.

It is also possible to include the trustee, stereotype, and context features in this reinterpretation model,

$$\rho_w(\cdot) = F(\forall \langle te, \theta_{te}, C, M, f_w^{ml}(I); f_{tr}^{ml}(I)\rangle \in \mathbf{I}_{tr} \cup \mathbf{I}w), \tag{33}$$

for when differences in agent perspectives are dependent on the situation. This can then be applied as,

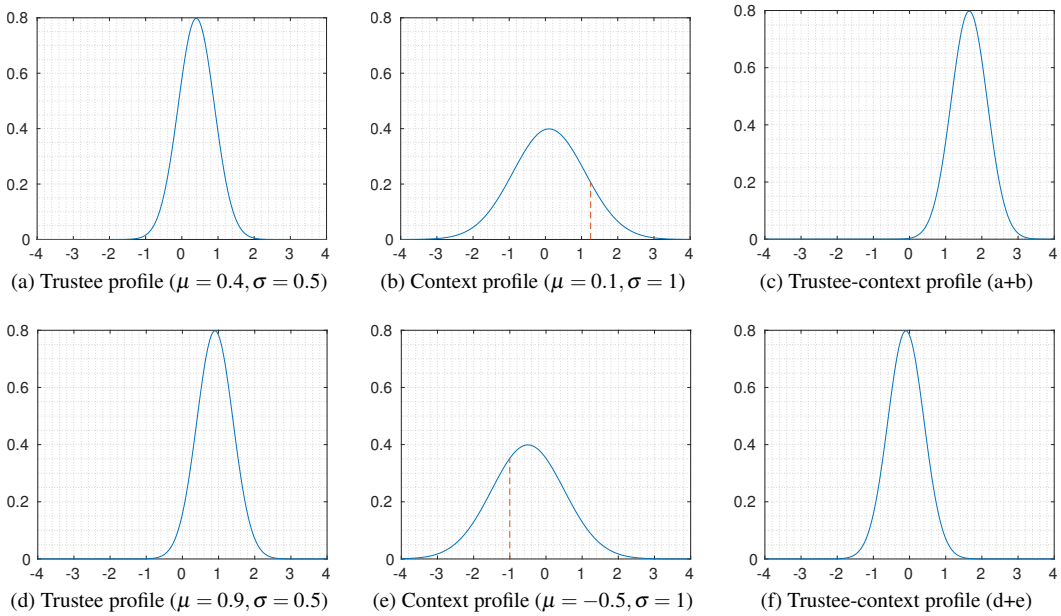
$$\rho(\langle te, \theta_{te}, C, M, f_w^{ml}(\langle te, \theta_{te}, C, M, \rangle)\rangle \in S) = \rho(S, f_w^{ml}(S)) \tag{34}$$

The final reputation value output is an aggregate of the direct interaction trust mdoel, $f_{tr}(\cdot)$, and reinterpreted outputs of the witness function, $f_w(\cdot)$. Many different aggregation functions can be used, but in this work we use a weighted mean of model outputs. Reputation is then computed as,

$$\mathscr{R}(S) = \frac{1}{|\mathbf{W}|+1}\left[\omega_{tr} f_{tr}^{ml}(S) + \sum_{w \in \mathbf{W}} \omega_w \rho_w(S, f_w^{ml}(S))\right], \tag{35}$$

and is referred to as the MLRS strategy.

Although MLRS requires $1 + 2|\mathbf{W}|$ models to be learned in total, each of the individual models can be relatively simple and are often able to be learned from small sample sizes. Naïve Bayes has a computational complexity of $O(nm)$, for example, which means that models can be learned quickly. The interactions used to learn each trustor and witness model are also only those interactions recorded by the trustor and provided by the individual witnesses, meaning that n is often not large. If the sample is too large, subsampling techniques can be applied to reduce its size, and many machine learing algorithms, including naïve Bayes, can be updated with new samples to avoid a full re-learning of the model. Also, features may be selected using feature selection techniques such as Mutual Information and Principal Components Analysis, to limit m and reduce learning time further [41].



(a) Trustee profile ($\mu = 0.4, \sigma = 0.5$) (b) Context profile ($\mu = 0.1, \sigma = 1$) (c) Trustee-context profile (a+b)

(d) Trustee profile ($\mu = 0.9, \sigma = 0.5$) (e) Context profile ($\mu = -0.5, \sigma = 1$) (f) Trustee-context profile (d+e)

Figure 1: Two example combinations of trustee and context profiles.

Trustee profile	μ_{te}	σ_{te}	θ_{te}
0	0.0	[0.0,0.1,...,0.5]	111110000000000
1	0.1	[0.0,0.1,...,0.5]	011111000000000
2	0.2	[0.0,0.1,...,0.5]	001111100000000
3	0.3	[0.0,0.1,...,0.5]	000111110000000
4	0.4	[0.0,0.1,...,0.5]	000011111000000
5	0.5	[0.0,0.1,...,0.5]	000001111100000
6	0.6	[0.0,0.1,...,0.5]	000000111110000
7	0.7	[0.0,0.1,...,0.5]	000000011111000
8	0.8	[0.0,0.1,...,0.5]	000000001111100
9	0.9	[0.0,0.1,...,0.5]	000000000111110
10	1.0	[0.0,0.1,...,0.5]	000000000011111

(a) Trustee profiles

C	μ_C	σ_C	θ_C
A	-0.5	1	111110000000000
B	-0.4	1	011111000000000
C	-0.3	1	001111100000000
D	-0.2	1	000111110000000
E	-0.1	1	000011111000000
F	0.0	1	000001111100000
G	0.1	1	000000111110000
H	0.2	1	000000011111000
I	0.3	1	000000001111100
J	0.4	1	000000000111110
K	0.5	1	000000000011111

(b) Context profiles

Table 3 List of (a) trustee and (b) context profiles, their base means and STDs, and their features.

6 Evaluation

To evaluate models in our abstraction we use a simulated marketplace based on that presented by Burnett et al. [6, 7], and subsequently adopted by Şensoy et al. [34] and Taylor et al. [35, 36]. The simulation and models discussed in this evaluation are available for download as open source software.[1] The simulation consists of trustor agents that assess the reputation of, and interact with, and trustee agents over 250 rounds. At the beginning of the simulation each trustee is assigned a base capability, μ_{te}, in the range $[0.0, 0.1, \ldots, 1.0]$, and a base STD, σ_{te}, in the range $[0.1, 0.2, \ldots, 0.5]$. Similarly, each interaction context is assigned a base mean, μ_C, in the range $[-0.5, -0.4, \ldots, 0.5]$ and a STD of $\sigma_C = 1$. The mean capability of a trustee, te, for a particular context, C, is given as,

$$\mu_{te,C} = \mu_{te} + sample(\mu_C, \sigma_C), \tag{36}$$

where $sample(\mu_C, \sigma_C)$ is a random sample drawn from the Gaussian distribution defined by μ_C and σ_C. The mean capability, $\mu_{te,C}$, and STD, σ_{te}, then define the Gaussian from which interaction outcomes with a trustee in a context are drawn, $sample(\sigma_{te,C}, \sigma_{te})$.

[1] github.com/jaspr-project/MLReputationTestBed

Two example combinations are shown in Figure 1. Figure 1(a) displays trustee profile 4, where $\mu_{te} = 0.4$, with a standard deviation of $\sigma_{te} = 0.5$, and Figure 1(b) shows context profile G. Their combination, having sampled $\mu_{context} = 1.25$ from the context profile distribution (indicated by the vertical dashed line on the context profile), is displayed in Figure 1(c). The mean of this combined trustee-context profile is $\mu_{te,C} = 1.65$, which is higher than the base trustee profile mean and represents that this trustee is more proficient in this context than on average. A second trustee profile, with $\mu_{te} = 0.9$ and $\sigma_{te} = 0.5$, is shown in Figure 1(d), and is combined with the context profile, with $\mu_C = -0.5$, shown in Figure 1(e). The sample taken from the context profile is -1, meaning that the trustee-context profile in Figure 1(e) has a mean of -0.1, representing that this trustee does not perform well in this context compared to the base trustee profile.

After each interaction with a trustee, the trustor then rates that interaction based on their preference threshold, which is assigned randomly at the beginning of the simulation in the range $[0.0, 0.1, \ldots, 1.0]$. An interaction with an outcome greater than their preference threshold is deemed successful and given a rating of 1 by the trustor, otherwise it is rated as 0. This means that a preference threshold of 0.5 leads to a random strategy having a successful interaction 50% of the time.

Each trustee is also assigned a set of observable stereotype traits, which are indicative of their base capability, as outlined in Table 3(a). The observable traits distinguish each of the profiles, to be used in stereotype assessments of trustees. Each element in these feature vectors can be interpreted as the trustee exhibiting a trait or not, e.g. the first trait may represent 'airport transfer'. Similarly, each interaction context is associated with a set of traits (outlined in Table 3(b)) for use in contextual reputation assessments. To introduce noise into stereotype and context traits, each of binary values are flipped (i.e. $0 \rightarrow 1$ and $1 \rightarrow 0$) with a probability. With a noise level of 0.25, for instance, each trait value is flipped with a probability of 0.25, whereas a noise level of 0 means that no values are changed. For stereotype traits the noise is applied when assigning them to a trustee at the beginning of the simulation. With context traits, the noise is applied separately for each trustor at the beginning of each round. The traits observed by the trustor in a round are used in their reputation assessment and stored in the subsequent interaction record.

Each trustor and trustee agent leaves the simulation in each round with a probability of 0.05, to be replaced by another agent. New trustees are assigned a base capability, context capabilities, and stereotype traits using the same process as described above. The number of agents in the simulation is static, therefore, and in all of our simulations there were 100 trustee agents and 20 trustor agents. In each round, each trustor agent is given a random 10 available trustees from which they select the one with highest reputation as an interaction partner. Similarly, in each reputation assessment, each trustor requests reputation information from 10 random witnesses.

Each trustor is either an honest, negative, random, or slanderous witness (which is determined when they join the simulation). Honest witnesses report all their reputation information truthfully, whereas negative witnesses report an opinion computed using the negation of all their ratings. For example, if they gave a rating of 0.4 for an interaction, they will use -0.4 as the rating when computing an opinion to be communicated to the requesting trustor. A slanderous witness reports true assessment of half of the trustees, and for the remainder they report an opinion computed using decreased ratings, specifically $0.5(r - 1)$. Finally, a random witness computes their opinions using random ratings for all their interactions, before communicating that opinion.

The evaluation is separated by the information used by the strategies. We first investigate basic trust, where only the trustee identifier is used by the strategy, and then introduce known context, stereotype, and unforeseen context features. Two instantiations are considered for each kind of strategy, namely one using an aggregation (as described in Section 3) and another based in machine learning (as described in Section 5).

The strategies based on aggregations are instantiated using probabilistic reputation models, namely BRS, TRAVOS, BLADE, and HABIT. The opinion representation used was therefore a Beta PDF, as in Equation 10, and the trust function computes its expected value, as in Equation 12. These models are used because of their probabilistic nature in predicting whether an interaction will be successful or not,

Strategy	Description	Representative models
RAND	Uniformly random reputation score regardless of input.	N/A
D, *MLD*	Direct information only.	N/A
DW, *MLDW*	Direct and witness information.	REGRET, BRS, FIRE, Burnett
DDW, *MLDDW*	Direct information and discounted witness information.	TRAVOS, STAGE
DRW, *MLDRW*	Direct information and reinterpreted witness information.	HABIT, BLADE, MLRS

Table 4 List of opinion aggregations investigated. Machine learning variants of the strategies are prefixed with *ML*.

Strategy	**One context**		**Five contexts**		**Ten contexts**	
RAND	0.524	(0.030)	0.526	(0.010)	0.499	(0.004)
D	0.730	(0.030)	0.564	(0.012)	0.543	(0.004)
DW	1.280	(0.020)	**0.835**	(0.012)	**0.724**	(0.005)
DDW	1.293	(0.022)	0.794	(0.011)	0.705	(0.005)
DRW (HABIT)	1.021	(0.025)	0.598	(0.011)	0.566	(0.004)
DRW (BLADE)	1.042	(0.022)	0.618	(0.011)	0.566	(0.005)
MLD	0.715	(0.031)	0.557	(0.012)	0.545	(0.004)
MLDW	**1.318**	(0.022)	0.813	(0.011)	0.714	(0.005)
MLDDW	1.250	(0.021)	0.792	(0.011)	0.717	(0.004)
MLDRW	1.052	(0.024)	0.615	(0.011)	0.569	(0.004)

Table 5 Utilities for strategies using basic trust information only, with different numbers of contexts and honest witnesses. The standard error is displayed in braces after the mean utility and bold signifies the highest mean utility gained by a strategy in the environment.

which can be implemented using machine learning as a binary classification task. For the machine learned reputation strategies, a random forest [5] was learned using the trustor's direct interaction records. A random forest model is made up of several decision trees, each learned using a random subset of samples (via bootstrap aggregating [4]) and a random subset of features (via random subspace learning [11]). Typically, a majority vote over the decision trees is used, where the modal prediction is taken as the overall output. In this paper we aggregate the likelihood outputs of each of the decision trees and take the mean likelihood of success as the trust score. This is then a probabilistic estimate of the rating, akin to that provided a Beta PDF.

The opinion aggregations investigated are outlined in Table 4. The aggregation strategy using direct information only, *D*, and the machine learning variant, *MLD*, both use the trustor's opinion only in computing the trust score. To combine direct and witness information, *DW* aggregates the opinions as in Equation 23, and *MLDW* aggregates the opinions using Equation 30. The discounting of unreliable witness information in *DDW* is as in Equation 25, and uses computes the witness weights as in TRAVOS [37]. *MLDDW* discounts unreliable witness information using Equation 31, for which the respective model accuracies, estimated using the direct interaction records, are the weights. To reinterpret witness information (*DRW*), we use both BLADE [29] (denoted *DRW* (HABIT)) and HABIT [38] (denoted *DRW* (BLADE)). In the machine learning case, MLRS is used, as in Equation 35.

All results presented are the mean utility gained by agents over the 250 simulation rounds, averaged over 100 simulations. To analyse the results we have performed pair-wise t-tests and corrected for multiple-comparisons using the Bonferroni correction. Each conclusion discussed is significant at the $p < 0.01$ level, unless otherwise stated.

6.1 Basic reputation

With basic reputation information, strategies do not account for any context or stereotype features. The aggregations used an interaction weighting as in Equation 9 and the machine learning models used only the $\langle te \rangle$ as an input feature. Table 5 shows the mean interaction utilities gained over 250 rounds in simulations where all witness agents were honest and with one, five, and ten interaction contexts selected at random from the contexts outlined in Table **??**. The random strategy, *RAND*, gained a mean utility of around 0.5 as expected, which was significantly worse than all other strategies in all cases. This indicates that assessing

Strategy	Single context		5 contexts		10 contexts	
D_{st}	0.687	(0.030)	0.685	(0.011)	0.673	(0.004)
DW_{st}	0.933	(0.034)	0.892	(0.012)	0.895	(0.005)
DDW_{st}	0.839	(0.033)	0.796	(0.012)	0.784	(0.005)
$DRW(HABIT)_{st}$	0.743	(0.028)	0.708	(0.011)	0.700	(0.004)
$DRW(BLADE)_{st}$	0.723	(0.028)	0.718	(0.010)	0.702	(0.004)
D_{st} (Burnett)	0.793	(0.031)	0.630	(0.013)	0.615	(0.004)
DW_{st} (Burnett)	**1.343**	(0.022)	0.947	(0.010)	0.875	(0.004)
MLD_{st}	0.789	(0.030)	0.642	(0.012)	0.630	(0.004)
$MLDW_{st}$	1.227	(0.028)	**0.974**	(0.012)	**0.903**	(0.005)
$MLDDW_{st}$	1.275	(0.025)	0.946	(0.011)	0.891	(0.004)
$MLDRW_{st}$	0.959	(0.030)	0.730	(0.012)	0.699	(0.005)

Table 6 Utilities for strategies using stereotype information, with different numbers of contexts and honest witnesses. The standard error is displayed in braces after the mean utility and bold signifies the highest mean utility gained by a strategy in the environment.

reputation with information from previous interactions is beneficial, even if only direct interactions are used (as in *D* and *MLD*). In all cases, lower utilities were gained in simulations with more contexts.

Using witness information provided significantly higher utilities than using direct information only. Discounting witness information, in *DDW* and *MLDDW*, gained similar utilities to using witness information directly, in *DW* and *MLDW*. However, reinterpreting witness reports in *DRW* (HABIT), *DRW* (BLADE), and *MLDRW*, provided significantly worse utilities than these strategies. This indicates that when witnesses are known to be honest there is no benefit to discounting their reports, and it is detrimental to attempt to translate them to the perspective of the trustor.

In general, the utilities gained by the machine learned strategies were not significantly different to the respective aggregation strategy. For instance, there was no significant difference between the utilities gained by *MLDW* and *DW* for any number of contexts. The same was true for the all other opinion aggregations, showing that machine learning can be used to replicate aggregation strategies.

6.2 Stereotypes

Stereotype information was included in the aggregation models using the interaction weighting function in Equation 16. The utilities in Table 6 show that such aggregation strategies performed very poorly compared to the machine learning models, which used stereotype traits as binary input features. In simulations with a single context, all aggregation models gained lower utilities when using stereotype information than when using basic information only. With multiple contexts their performances were better with stereotype information, but in most cases were still worse than the performances of the machine learning strategies in all cases. Another kind of stereotype reputation, proposed by Burnett et al. [6, 7] and defined by Equation 18, is therefore included in this table. Although the Burnett model used machine learning for the stereotype model, it represents opinions using Beta PDFs and uses direct and witness information in the same way as *DW* and *MLDW*.

Using direct information only, as with basic trust, generally gave significantly lower utilities than using both direct and witness information. One exception was the reinterpretation strategies, *DRW* (HABIT) and *DRW* (BLADE), which gained slightly lower utilities than *D*. Furthermore, the utilities gained by these strategies in simulations with single contexts was not significantly different to those gained when using basic information. In multi-context simulations utilities when using stereotype information were generally higher compared to not using stereotypes.

By far the highest utilities were gained by strategies that used machine learning with both direct and witness information. With any number of contexts in the simulation, Burnett's strategy, *MLDW*, and *MLDDW* gained significantly the highest utilities. Although there was no statistically significant difference between the three strategies in any case, Burnett's strategy did gain the highest utility in simulations with a single context. With ten contexts in the simulation, *MLDW* and *MLDDW* outperformed Burnett's strategy by a slight margin, as did *DW*.

Strategy	Single context		5 contexts		10 contexts	
D_{ctx}	0.726	(0.032)	0.584	(0.011)	0.538	(0.004)
DW_{ctx}	**1.281**	(0.023)	**0.908**	(0.010)	**0.789**	(0.004)
DDW_{ctx}	1.261	(0.023)	0.893	(0.011)	0.756	(0.005)
$DRW(HABIT)_{ctx}$	1.058	(0.025)	0.619	(0.012)	0.552	(0.004)
$DRW(BLADE)_{ctx}$	0.730	(0.031)	0.571	(0.011)	0.556	(0.004)
MLD_{ctx}	0.705	(0.029)	0.553	(0.011)	0.546	(0.004)
$MLDW_{ctx}$	1.279	(0.021)	0.868	(0.010)	0.762	(0.005)
$MLDDW_{ctx}$	1.263	(0.023)	0.855	(0.010)	0.734	(0.005)
$MLDRW_{ctx}$	1.079	(0.023)	0.674	(0.011)	0.617	(0.005)

Table 7 Utilities for strategies using known context information, with different numbers of contexts and honest witnesses. The standard error is displayed in braces after the mean utility and bold signifies the highest mean utility gained by a strategy in the environment.

6.3 Known-contexts

To incorporate context information into the aggregation strategies, the interaction weighting in Equation 19 was used, using Jaccard similarity between the context traits. For the machine learning approaches, the binary feature vector indicating which context traits were observed was used as inputs to the learned models. It is assumed that all agents represent context in the same way, but this does not need to be true in general [35]. The results for using known context information are shown in Table 7.

As with basic trust, the strategies using direct information only tended to gain significantly lower utilities than those that used both direct and witness information. One exception to this was *DRW* (HABIT) and *DRW* (BLADE), which gained similar utilities to *D* and *MLD* when there were multiple contexts in the simulation. *MLDRW*, while gaining lower utilities than strategies that did not reinterpret witness opinions, did gain higher utilities than *DRW* strategies. Furthermore, *D*, *MLD*, *DRW* (HABIT), and *DRW* (BLADE), did not improve on their performances compared to when context information was not used, and *MLDRW* gained only slightly more utility in simulations with multiple contexts. One explanation for this is that trustor agents were unable to individually collect enough interaction records to discriminate between contexts themselves, and as a result could not reinterpret contextual opinions from witnesses.

For any number of simulation contexts, no strategy gained significantly lower utilities when using context information than when using basic information only. Furthermore, strategies using context information did not achieve significantly higher utilities than strategies using basic information. This means that, for an unknown number of contexts, there is no cost in utility to using context information when assessing trust and reputation, but also that context information has no benefit when there only is only one context. In simulations with multiple contexts, all strategies gained lower utilities. However, strategies that used witness information without reinterpretation, namely *DW*, *MLDW*, *DDW*, and *MLDDW*, gained significantly more utility when using context information compared to using basic information.

Of the four best performing strategies, *DW* gained the most utility by a small but statistically significant margin in simulations with ten contexts, outperforming *MLDW* by around 0.027. Similarly, *DDW* gained about 0.022 more in utility than *MLDDW*. These differences are possibly due to the difference in the two aggregation approaches, and the limited number of samples available for the learning algorithm to discriminate between the contexts. An alternative learning approach that improves on this performance is discussed in Section 6.7, where a single model is learned using both direct and witness interaction records.

6.4 Unforeseen context

Table 8 shows utilities gained in simulations where interactions encountered unforeseen contexts with different likelihoods, for a single known context. The utility of interactions with unforeseen contexts was always -1. As a result, with higher likelihoods of unforeseen contexts the utilities gained by all strategies was lower. To mitigate for ratings made after an interaction with an unforeseen context, strategies used the interaction weighting function defined in Equation 22. In the machine learning strategies this weighting was applied to the training samples prior to learning.

Strategy	0.0		0.05		0.1		0.25	
RAND	0.524	(0.030)	0.418	(0.028)	0.390	(0.031)	0.102	(0.023)
D	0.730	(0.030)	0.621	(0.031)	0.563	(0.028)	0.271	(0.025)
DW	1.280	(0.020)	**1.200**	(0.021)	1.009	(0.021)	0.615	(0.019)
DDW	1.293	(0.022)	1.126	(0.023)	1.038	(0.020)	0.642	(0.017)
DRW (HABIT)	1.021	(0.025)	0.874	(0.024)	0.761	(0.023)	0.362	(0.020)
DRW (BLADE)	1.042	(0.022)	0.954	(0.023)	0.802	(0.022)	0.399	(0.020)
MLD	0.715	(0.031)	0.619	(0.030)	0.575	(0.025)	0.238	(0.023)
MLDW	**1.318**	(0.022)	1.173	(0.022)	0.986	(0.019)	0.637	(0.017)
MLDDW	1.250	(0.021)	1.119	(0.021)	0.961	(0.019)	0.626	(0.017)
MLDRW	1.052	(0.024)	0.896	(0.023)	0.769	(0.023)	0.379	(0.022)
D_{mit}	0.761	(0.030)	0.557	(0.029)	0.441	(0.027)	0.223	(0.022)
DW_{mit}	1.288	(0.022)	1.169	(0.023)	**1.061**	(0.020)	**0.700**	(0.017)
DDW_{mit}	1.270	(0.021)	1.169	(0.020)	1.049	(0.019)	0.643	(0.018)
$DRW(HABIT)_{mit}$	1.091	(0.026)	0.932	(0.025)	0.822	(0.023)	0.466	(0.021)
$DRW(BLADE)_{mit}$	1.048	(0.023)	0.963	(0.022)	0.834	(0.020)	0.474	(0.021)
MLD_{mit}	0.727	(0.030)	0.585	(0.030)	0.555	(0.028)	0.247	(0.023)
$MLDW_{mit}$	1.306	(0.024)	1.193	(0.022)	1.035	(0.021)	0.688	(0.017)
$MLDDW_{mit}$	1.233	(0.023)	1.132	(0.022)	0.984	(0.021)	0.675	(0.017)
$MLDRW_{mit}$	1.016	(0.023)	0.893	(0.022)	0.801	(0.022)	0.445	(0.020)

Table 8 Utilities for strategies using unforeseen context information, with different likelihoods of unforeseen contexts and honest witnesses. The standard error is displayed in braces after the mean utility and bold signifies the highest mean utility gained by a strategy in the environment.

Strategy	50% Negation		50% Random		50% Slander	
RAND	0.506	(0.033)	0.474	(0.032)	0.494	(0.030)
D	0.655	(0.030)	0.671	(0.029)	0.676	(0.033)
DW	0.678	(0.032)	1.044	(0.026)	1.330	(0.025)
DDW	0.799	(0.028)	**1.140**	(0.022)	1.322	(0.024)
DRW (HABIT)	1.061	(0.025)	0.895	(0.026)	1.010	(0.025)
DRW (BLADE)	**1.080**	(0.023)	0.971	(0.025)	1.068	(0.024)
MLD	0.681	(0.031)	0.703	(0.031)	0.639	(0.031)
MLDW	0.567	(0.028)	1.022	(0.027)	**1.363**	(0.025)
MLDDW	0.494	(0.032)	1.012	(0.023)	1.271	(0.023)
MLDRW	0.984	(0.025)	0.883	(0.028)	1.062	(0.022)

Table 9 Utilities for strategies using basic trust information, when half of all witnesses are either negation, random, or slander. The standard error is displayed in braces after the mean utility and bold signifies the highest mean utility gained by a strategy in the environment.

Regardless of mitigation, using direct information only in *D* and *MLD* or reinterpreting witness information in *DRW* (BLADE), *DRW* (HABIT), and *MLDRW*, provided significantly lower utilities. In general, the difference in utility gained was not significant when witness information was used without reinterpretation. For higher likelihoods the mitigation strategies, *DW*, *DDW*, and *MLDW* outperformed the respective non-mitigation strategy by a small but not statistically significant margin. The machine learning strategies, therefore did not perform significantly differently to their aggregation counterparts.

6.5 Dishonesty

Table 9 shows utilities gained by strategies when 50% of witness agents either negate their ratings when reporting opinions, use random ratings, or slander trustees. In each case only basic trust information is used, and there is only one simulation context. When direct information is used in *D* and *MLD* there was no significant difference in utilities from when all witnesses are reliable. Using witness information directly, in *DW* and *MLDW*, the utility was significantly lower when there were either negation or random witnesses, compared to with all honest witnesses. They were not, however, significantly different with slander witnesses.

With negation witnesses, *DRW* (HABIT), *DRW* (BLADE) and *MLDRW* gained significantly highest utilities, as they were able to reinterpret the witness opinions properly. These utilities were also not significantly different to those that the reinterpretation strategies gained with no unreliable witnesses. *MLDDW* performed particularly poorly in this case, compared to *DDW*. *DDW* also gained the highest utilities with random witnesses, but was not statistically significantly better than *MLDW* or *MLDDW*.

Strategy	Basic		st + ctx		ctx + mit		st + mit		st + ctx + mit	
RAND	0.425	(0.004)		NA		NA		NA		NA
D	0.459	(0.004)	0.566	(0.004)	0.464	(0.004)	0.585	(0.004)	0.571	(0.004)
DW	**0.629**	(0.005)	0.735	(0.006)	0.692	(0.004)	0.798	(0.005)	0.735	(0.005)
DDW	0.616	(0.004)	0.673	(0.005)	**0.670**	(0.004)	0.691	(0.005)	0.689	(0.005)
DRW (HABIT)	0.481	(0.004)	0.575	(0.004)	0.486	(0.004)	0.615	(0.004)	0.581	(0.004)
DRW (BLADE)	0.493	(0.004)	0.606	(0.004)	0.485	(0.004)	0.620	(0.004)	0.613	(0.005)
MLD	0.458	(0.004)	0.545	(0.004)	0.469	(0.004)	0.545	(0.004)	0.547	(0.004)
MLDW	0.614	(0.005)	**0.825**	(0.005)	0.642	(0.004)	**0.806**	(0.005)	**0.828**	(0.005)
MLDDW	0.616	(0.004)	**0.825**	(0.005)	0.653	(0.004)	0.803	(0.004)	0.820	(0.004)
MLDRW	0.489	(0.004)	0.649	(0.004)	0.531	(0.004)	0.606	(0.004)	0.655	(0.005)

(a) Honest witnesses

Strategy	Basic		st + ctx		ctx + mit		st + mit		st + ctx + mit	
RAND	0.419	(0.003)		NA		NA		NA		NA
D	0.463	(0.003)	0.574	(0.004)	0.462	(0.004)	0.587	(0.004)	0.573	(0.004)
DW	0.492	(0.004)	0.554	(0.006)	0.511	(0.005)	0.600	(0.007)	0.562	(0.006)
DDW	**0.506**	(0.004)	0.597	(0.004)	**0.514**	(0.004)	**0.627**	(0.004)	0.597	(0.005)
DRW (HABIT)	0.481	(0.004)	0.567	(0.004)	0.484	(0.004)	0.608	(0.004)	0.577	(0.004)
DRW (BLADE)	0.479	(0.004)	**0.609**	(0.004)	0.479	(0.004)	0.617	(0.004)	**0.613**	(0.005)
MLD	0.472	(0.004)	0.546	(0.004)	0.455	(0.004)	0.542	(0.004)	0.553	(0.003)
MLDW	0.477	(0.005)	0.587	(0.007)	0.497	(0.004)	0.584	(0.007)	0.582	(0.007)
MLDDW	0.500	(0.005)	0.572	(0.006)	0.503	(0.005)	0.576	(0.007)	0.577	(0.006)
MLDRW	0.474	(0.004)	0.604	(0.005)	0.510	(0.004)	0.568	(0.005)	0.609	(0.004)

(b) Dishonest witnesses

Table 10 Utilities for strategies using different combinations of trust information with (a) honest witnesses and (b) witnesses that are each 25% likely to be either random, slander, or negation. In each simulation there were ten contexts and freak events occurred with a likelihood of 0.05. The standard error is displayed in braces after the mean utility and bold signifies the highest mean utility gained by a strategy in the environment.

With slander witnesses there were no significant differences in utility gained by any of the strategies when compared to when all witnesses were reliable.

6.6 Putting it all together

When combining multiple kinds of trust information, such as stereotype and context, in one model, aggregation approaches must resort to combining the respective interaction weights. In these results we combine weights by taking their product, e.g. stereotype and context information is combined by multiplying their respective interaction weights, $\Psi_{st+ctx}(S,I) = \Psi_{st}(S,I) \times \Psi_{ctx}(S,I)$. For machine learning strategies, the features are simply appended together and all of them are used in the learned model.

Utilities gained by strategies using different combinations of stereotype, known-context and unknown-context, information are shown in Table 10. The simulations all contained ten contexts and freak events had a likelihood of 0.05. Witnesses for results in Table 10(a) were all honest, and results in Table 10(b) were from simulations with dishonest witnesses that were each 25% likely to be either random, slander, or negation.

Using basic information only, machine learning strategies had similar performance to the aggregation strategies. With honest witnesses, all strategies except those that reinterpreted witness reports gained similar higher utilities, whereas *DRW* (HABIT), *DRW* (BLADE), and *MLDRW*, gained utilities close to that of *RAND*. In the presence of dishonest witnesses, all strategies using basic information only performed poorly.

Using stereotype and known-context information ($st + ctx$) increased performance in all cases with honest witnesses. In this environment the machine learning strategies gained significantly higher utilities than did their aggregation counterparts. This was also the case when stereotype and unknown context information was combined ($st + mit$), and also when all three kinds of information were combined ($st + ctx + mit$). When known-context and unknown-context information ($ctx + mit$) were combined, the aggregation strategies gained slightly higher utilities than the machine learning strategies. Overall, this shows that machine learning methods are more equipped to use these different combinations of

Strategy	One context		Five contexts		Ten contexts	
DW	1.280	(0.020)	0.835	(0.012)	0.724	(0.005)
DW_{ctx}	1.281	(0.023)	0.908	(0.010)	0.789	(0.004)
MLDW	**1.308**	(0.022)	0.793	(0.011)	0.697	(0.005)
$MLDW_{ctx}$	1.280	(0.021)	**0.963**	(0.010)	**0.818**	(0.004)

Table 11 Utilities for *DW* and *MLDW* (using an alternative opinion aggregation) with and without context information, and in simulations with different numbers of contexts and honest witnesses. The standard error is displayed in braces after the mean utility and bold signifies the highest mean utility gained by a strategy in the environment.

Strategy	One context		Five contexts		Ten contexts	
DW	1.462	(0.026)	0.862	(0.010)	0.752	(0.005)
MLDW (Eq 24)	1.468	(0.031)	0.860	(0.012)	0.737	(0.005)
MLDW (Eq 23)	**1.519**	(0.034)	0.842	(0.013)	0.738	(0.005)
DW_{ctx}	1.388	(0.026)	0.945	(0.012)	0.805	(0.005)
$MLDW_{ctx}$ (Eq 24)	1.444	(0.028)	0.908	(0.010)	0.791	(0.005)
$MLDW_{ctx}$ (Eq 23)	1.521	(0.035)	**1.013**	(0.011)	**0.851**	(0.005)

Table 12 Utilities for *DW* using continuous ratings and *MLDW* using a multi-dimensional learning algorithm, both with and without context information. The simulations had different numbers of contexts and honest witnesses. The standard error is displayed in braces after the mean utility and bold signifies the highest mean utility gained by a strategy in the environment.

features. Further, the combination of stereotype and known-context information ($st + ctx$) gained the highest utilities, when used in the *MLDW* and *MLDDW* strategies. The addition of unknown-context information to these models did not improve performance significantly.

With dishonest witnesses there was generally no significant difference between the machine learning and aggregation strategies. Some exceptions to this are with stereotype and unknown-context information ($st + mit$), where *DDW* significantly outperformed *MLDDW*, and with known-context and unknown-context information ($ctx + mit$) where *MLDRW* outperformed both *DRW* (HABIT) and *DRW* (BLADE). These observations may indicate that, in general, dishonest witnesses cancel out the benefits of applying machine learning to combinations of features. When using dishonest witnesses as information sources, the inclusion of stereotype information in any model provided the highest utilities, regardless of what it was combined with. For instance, $st + ctx$ had similar results to $\theta + mit$ and $\theta + ctx + mit$, indicating that stereotypes are the most important information to use in models in the presence of dishonesty.

6.7 Single learned model

The aggregation of the machine learned opinions produced lower utilities than the aggregation used in *DW*. In this section, therefore, we use an alternative aggregation for machine learned opinions, where the models were combined rather than the outputs. This is done in a simple way, by concatenating the training samples and learning a single random forest model. Using this approach it is difficult to discount or reinterpret the opinions, and so only results *DW* and *MLDW* are presented. Alternative methods may exist where model parameters are combined, weighted, or reinterpreted, but this is left as future work.

Table 11 shows utilities in simulations with one, five, and ten contexts, for the *DW* and *MLDW* when using both basic and context trust information. With a single context, the two strategies gained similar utilities regardless of whether context information was used or not. In simulations with multiple contexts, *MLDW* and *DW* also did not have significantly different utilities when they did not use context information. When using context information with multiple contexts, however, *MLDW* gained significantly more utility than *DW*.

6.8 Continuous ratings

In all the results presented thus far, interaction ratings have been been binary and reputation has been probabilistic. Specifically, ratings of interactions have been positive where the outcome was above the trustor's preference threshold, and negative otherwise. Higher utilities can be gained by reputation strategies that use continuous ratings, in particular ratings that are computed as the difference between the interaction utility and preference threshold. The FIRE reputation system uses the interaction aggregation defined in Equation 8, and uses continuous ratings to output a continuous trust and reputation scores.

To mirror this in machine learning reputation, either a regression learning algorithm or multi-label classifier can be used. In this paper we simply increase the dimensionality of the target variable in the random forest algorithm, to make a multi-label classifier. In particular, rather than mapping input features to two values, the target can now take one of five values that span the intervals $(-\infty : -1.5), [-1.5 : -0.5), [-0.5 : 0.5), [0.5 : 1.5), [1.5 : \infty)$.

Table 12 shows results for these variants of the *DW* and *MLDW* reputation systems in simulations with one, five, and ten contexts. The first *MLDW* is based on the aggregation of outputs, and is the same reputation system as outlined in Sections 3.2.1 and 3.2.4. The second *MLDW* uses the aggregation presented in Section 6.7, and concatenates the training samples prior to learning. With any number of contexts in the simulation, the utilities gained by these strategies were significantly higher than the binary strategies using the same information. When using basic trust information only, there was no significant difference between the utilities gained between *DW* and *MLDW*. With context information, however, *MLDW* had significantly higher performance in simulations with multiple contexts than did *DW*. This again highlights the power of using machine learning as the basis of trust and reputation assessment.

7 Summary

In general, machine learning strategies had similar performance to the aggregation models using the same information sources. There were some situations where machine learning was better able to model features, in particular when stereotype information was used. When using context information, the aggregation strategies gained higher utilities than did the machine learning strategies that learned a model for each witness. When a single machine learning model was learned, by combining the data from all witnesses, however, the utility gained was significantly higher than for other strategies investigated. This indicates that the machine learning approach is limited by the number of samples available, which is overcome when direct and witness information are incorporated into the same training data for a single model. Methods for weighting training samples by the reliability of the witness, or reinterpreting them for different perspectives, should therefore be investigated to maximise the utility gained by single model variants of the *MLDDW* and *MLDRW* strategies.

8 Conclusion

In this paper we have reviewed the literature on the representation and dissemination of trust and reputation information. In analysing the representation we found that trust opinions are often represented as using PDFs formed via aggregations of previous ratings. Other trust and reputation systems represent trust information in models learned using machine learning over interaction tuples. Typically, reputation information is disseminated through a central authority or to acquaintances in a trust network. In either case, dishonesty of witness agents can be handled by discounting their opinions before combining them with the trustor's own. In cases where witness agents may have opinions that are different to those of the trustor in a consistent way, reinterpretations can be learned.

Following this review, a unifying abstraction of reputation systems for reputation systems was presented as opinion gathering, representation, and aggregation. As well as create instantiations of the abstraction to represent several existing reputation systems, we also presented instantiations based on machine learning over interaction tuples. A machine learning model was presented for each kind of reputation system found in the literature, to use the same kind of information in forming an opinion and aggregate opinions in a similar way. In culmination, the MLRS was presented to combine

these capabilities in one reputation system, including contextual reputation, stereotypes, and opinion reinterpretation.

Finally, an evaluation of the models discussed throughout the paper was performed using a simulated marketplace. The main findings from the evaluation were that the machine learning based strategies perform equally well to the aggregation based strategies using the same information sources. It was also clear that combining multiple information sources was more beneficial when using machine learning than weighted aggregations. For both the machine learning an aggregation models, however, using information relevant to the environment produced higher reputation assessments. For example, in an environment with multiple contexts it was beneficial to use context information, and in dynamic environments stereotype information was useful. The code for the simulations can be found as open source software.

As future work we intend to investigate combining model parameters, rather than their outputs, in combining opinions. This will enable more detailed reinterpretations of witness opinions, and possibly allow detection of dishonesty by inspecting the model for inconsistencies. Another avenue for research is to determine the usefulness of continuous, over discrete or binary, ratings in different kinds of environment.

References

[1] Abdullah M Aref and Thomas T Tran. A decentralized trustworthiness estimation model for open, multiagent systems (DTMAS). *Journal of Trust Management*, 2(1):1, 2015.

[2] Donovan Artz and Yolanda Gil. A survey of trust in computer science and the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):58–71, 2007.

[3] Ronald Ashri, Sarvapali D Ramchurn, Jordi Sabater, Michael Luck, and Nicholas R Jennings. Trust evaluation through relationship analysis. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1005–1011. ACM, 2005.

[4] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[5] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[6] Chris Burnett, Timothy J Norman, and Katia Sycara. Bootstrapping trust evaluations through stereotypes. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, pages 241–248, 2010.

[7] Chris Burnett, Timothy Norman, and Katia Sycara. Stereotypical trust and bias in dynamic multiagent systems. *ACM Transactions on Intelligent Systems and Technology*, 4(2):26, 2013.

[8] H. Fang, J. Zhang, M. Şensoy, and N. M. Thalmann. A generalized stereotypical trust model. In *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, pages 698–705, 2012.

[9] Nathan Griffiths. Enhancing peer-to-peer collaboration using trust. *Expert Systems with Applications*, 31(4):849–858, 2006.

[10] Ferry Hendrikx, Kris Bubendorfer, and Ryan Chard. Reputation systems: A survey and taxonomy. *Journal of Parallel and Distributed Computing*, 75:184–197, 2015.

[11] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.

[12] Bruno W. P. Hoelz and Célia G. Ralha. Towards a cognitive meta-model for adaptive trust and reputation in open multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 29(6):1125–1156, 2015.

[13] Trung Dong Huynh, Nicholas R Jennings, and Nigel R Shadbolt. FIRE: An integrated trust and reputation model for open multi-agent systems. In *16th European Conference on Artificial Intelligence*, pages 18–22, 2004.

[14] Trung Dong Huynh, Nicholas R Jennings, and Nigel R Shadbolt. An integrated trust and reputation model for open multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 13(2):119–154, 2006.

[15] Audun Jøsang and Roslan Ismail. The Beta reputation system. In *Proceedings of the 15th Bled Electronic Commerce Conference*, pages 41–55, 2002.

[16] Audun Jøsang, Roslan Ismail, and Colin Boyd. A survey of trust and reputation systems for online service provision. *Decision support systems*, 43(2):618–644, 2007.

[17] Andrew Koster, Jordi Sabater-Mir, and Marco Schorlemmer. Personalizing communication about trust. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, pages 517–524, 2012.

[18] Andrew Koster, Marco Schorlemmer, and Jordi Sabater-Mir. Engineering trust alignment: Theory, method and experimentation. *International Journal of Human-Computer Studies*, 70:450–473, 2012.

[19] X. Liu, T. Kaszuba, R. Nielek, A. Datta, and A. Wierzbicki. Using stereotypes to identify risky transactions in internet auctions. In *2010 IEEE Second International Conference on Social Computing*, pages 513–520, 2010.

[20] Xin Liu and Anwitaman Datta. Modeling context aware dynamic trust using hidden markov model. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, AAAI'12, pages 1938–1944, 2012.

[21] Xin Liu, Anwitaman Datta, Krzysztof Rzadca, and Ee-Peng Lim. Stereotrust: A group based personalized trust model. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 7–16, New York, NY, USA, 2009. ACM.

[22] Xin Liu, Anwitaman Datta, and Krzysztof Rzadca. Trust beyond reputation: A computational trust model based on stereotypes. *Electronic Commerce Research and Applications*, 12:24–39, 2013.

[23] Xin Liu, Gilles Tredan, and Anwitaman Datta. A generic trust framework for large-scale open systems using machine learning. *Computational Intelligence*, 30(4):700–721, 2014.

[24] Stephen Marsh. *Trust in distributed artificial intelligence*, pages 94–112. Springer Berlin Heidelberg, Berlin, Heidelberg, 1994.

[25] Simon Miles and Nathan Griffiths. Incorporating mitigating circumstances into reputation assessment. In *Proceedings of the 2nd International Workshop on Multiagent Foundations of Social Computing*, 2015.

[26] Simon Miles and Nathan Griffiths. Accounting for circumstances in reputation assessment. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems*, pages 1653–1654, 2015.

[27] Tim Muller, Yang Liu, and Jie Zhang. The fallacy of endogenous discounting of trust recommendations. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 563–572, 2015.

[28] Ross J. Quinlan. Learning with continuous classes. In *5th Australian Joint Conference on Artificial Intelligence*, pages 343–348, Singapore, 1992. World Scientific.

[29] Kevin Regan, Pascal Poupart, and Robin Cohen. Bayesian reputation modeling in e-marketplaces sensitive to subjectivity, deception and change. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1206–1212, 2006.

[30] Jordi Sabater and Carles Sierra. Regret: A reputation model for gregarious societies. In *4th Workshop on Deception Fraud and Trust in Agent Societies*, pages 61–69, 2001.

[31] Jordi Sabater and Carles Sierra. Social regret, a reputation model based on social relations. *ACM SIGecom Exchanges*, 3(1):44–56, 2001.

[32] Michael Schillo, Petra Funk, and Michael Rovatsos. Using trust for detecting deceitful agents in artificial societies. *Applied Artificial Intelligence*, 14(8):825–848, 2000.

[33] Murat Şensoy, Jie Zhang, Pinar Yolum, and Robin Cohen. Poyraz: Context-aware service selection under deception. *Computational Intelligence*, 25(4):335–366, 2009.

[34] Murat Şensoy, Burcu Yilmaz, and Timothy J. Norman. STAGE: Stereotypical trust assessment through graph extraction. *Computational Intelligence*, 32(1):72–101, 2016.

[35] Phillip Taylor, Nathan Griffiths, Lina Barakat, and Simon Miles. Stereotype reputation with limited observability. In *19th International Workshop on Trust in Agent Societies*, 2017.

[36] Phillip Taylor, Nathan Griffiths, Lina Barakat, and Simon Miles. Bootstrapping trust with partial and subjective observability. In *Proceedings of the 16th Conference on Autonomous Agents and Multiagent Systems*, 2017.

[37] WT Luke Teacy, Jigar Patel, Nicholas R Jennings, and Michael Luck. Coping with inaccurate reputation sources: Experimental analysis of a probabilistic trust model. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 997–1004, 2005.

[38] WT Luke Teacy, Michael Luck, Alex Rogers, and Nicholas R Jennings. An efficient and versatile approach to trust and reputation using hierarchical Bayesian modelling. *Artificial Intelligence*, 193: 149–185, 2012.

[39] Joana Urbano, Ana Paula Rocha, and Eugénio C Oliveira. Refining the trustworthiness assessment of suppliers through extraction of stereotypes. In *Proceedings of the International Conference on Enterprise Information Systems*, pages 85–92, 2010.

[40] Andrew Whitby, Audun Jøsang, and Jadwiga Indulska. Filtering out unfair ratings in bayesian reputation systems. In *Proceedings of the 7th International Workshop on Trust in Agent Societies*, volume 6, pages 106–117, 2004.

[41] Ian Witten, Eibe Frank, and Mark Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2011.

[42] Michael Wooldridge. *An introduction to multiagent systems*. John Wiley & Sons, 2009.

[43] Han Yu, Zhiqi Shen, Cyril Leung, Chunyan Miao, and Victor Lesser. A survey of multi-agent trust management systems. *IEEE Access*, 1:35–50, 2013.

[44] Giorgos Zacharia and Pattie Maes. Trust management through reputation mechanisms. *Applied Artificial Intelligence*, 14(9):881–907, 2000.