PRep: A Probabilistic Reputation Model for Biased Societies

Yasaman Haghpanah
University of Maryland Baltimore County
1000 Hilltop Circle, Baltimore, MD 21250
yasamanhj@umbc.edu

Marie desJardins
University of Maryland Baltimore County
1000 Hilltop Circle, Baltimore, MD 21250
mariedj@umbc.edu

ABSTRACT

Several reputation models have been introduced to deal with the problem of biased reputation providers. Most of these models discount or discard biased information received from the reputation providers, and most of them are not appropriate when a large population of information providers are biased or dishonest. In this paper, we present a probabilistic approach for reputation modeling, the Probabilistic Reputation model (PRep). PRep models a reputation provider's behavior, and uses this model to re-interpret the reported information, thus making use of the entire reputation reports effectively, even if they are biased. The re-interpreted data is combined with the agent's direct experiences to determine an overall level of trust in the third-party agent. We show that PRep significantly outperforms two state-of-the-art trust and reputation models—HAPTIC and TRAVOS—and improves the overall payoff in a game-theoretic environment.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: multiagent systems

General Terms

Human Factors, Design, Experimentation

Keywords

Reputation, Trust, Bayesian learning, Behavioral modeling

1. INTRODUCTION

Researchers have used reputation to model the trustworthiness of individuals in online markets, such as eBay, Amazon, and Yahoo [2, 4, 7]. eBay's tremendous success as an online auction site stems largely from its powerful yet simple reputation system, Feedback Forum [7]. The importance of reputation systems in Internet-mediated service provision has been widely recognized by researchers in various disciplines, such as multi-agent systems, economics, and information systems [4].

In the literature, reputation has been referred mostly to the aggregation of people's opinion about one person. In this paper, we use reputation as the perception of one person (or agent) about another person's behavior, intention, or reliability of service. This perception depends directly on the reputation reporter's characteristics, such as their level of uncertainty, whether they are biased or realistic, and/or cultural biases they may have. Different reputation characteristics can be dominant in specific domains. For example, the buyers' behavior in eBay can be modeled as biased towards giving positive or negative reviews. In reality, eBay's feedback forum has been observed to be surprisingly positive: among all ratings provided in eBay's feedback forum, 99% are positive [2, 7].

Avoiding unfair ratings while obtaining unbiased and honest reviews and ratings has been shown to be problematic and extremely hard to achieve [2]. Researchers in this area have explored different solutions to this problem. Some have tried to solve it by identifying unbiased reviews and using models that discount or discard the biased information [10, 12, 13]. Another proposed approach is to define a measure of review helpfulness, and identify the helpful reviews among a set of candidate reviews [3]. These approaches help to reduce the effect of biased and non-realistic reviews, and therefore highlight unbiased information that can be used for decision making. However, these proposed models are also throwing away data by filtering, discounting, and discarding, despite the fact that reviews are costly and in general not easily obtainable. Additionally, some products have few reviews, providing too little data to identify the fair reviews and discount the rest [3].

We propose the Probabilistic Reputation (PRep) model, a novel solution grounded in probabilistic modeling that learns the reviewers' behavior using Bayesian learning and then adjusts their reviews or ratings, as opposed to finding the unbiased reviews and discarding the rest. In the PRep framework, an agent first gathers information about a target agent through both direct interactions with that target and a reviewer's report about the target. Then, it learns the reporting agent's behavior by comparing these two sources (i.e., reports and direct experiences). After the learning phase is complete, the PRep agent can interpret other reports about other targets coming from the same report provider. As a result of this interpretation, it uses the entire report effectively, even if the report provider is biased (i.e., even if its reports are based on faulty perceptions or on dishonest reporting).

The key benefits of PRep are:

- The PRep reputation mechanism uses biased information as well as unbiased information; it therefore benefits from all available data.
- PRep agents obtain a tailored view of the reviewer (or reporter) according to their own behavior and preferences, resulting in customized aggregation of reviews.
- PRep is still effective in cases with very few observations or reviews. Most current models are unable to find usable feedback or generate a meaningful reputation level when only a few ratings are available [3].

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

In this paper, we describe our approach and its application in a game-theoretic environment. Our experimental results show that PRep is able to learn the reporting behavior of a report provider, and consequently to interpret other reports of that provider, resulting in better decision making and higher payoffs in its future interactions. Also, our results show that PRep identifies other agents' trustworthiness faster and more accurately than two other state-of-the-art trust and reputation models (TRAVOS and HAPTIC), even when reported information is biased.

2. RELATED WORK

Reputation has been widely studied [2, 4, 7, 8]. Several reputation models and mechanisms have been proposed in the literature to deal with the problem of biased and unfair ratings.

The BRS [13] and TRAVOS [10] approaches construct Bayesian models, using the number of satisfactory and unsatisfactory interactions with the sellers as ratings, and then use outlier detection or relevance analysis to filter out unreliable ratings. A drawback of these approaches is that a significant amount of information may be considered unreliable, and therefore discarded or discounted. BLADE [6] uses a Bayesian model reputation framework. In contrast to BRS and TRAVOS, it does not discard all unreliable ratings; rather, it learns an evaluation function for advisors who provide ratings close to their direct experience. Therefore, BLADE only performs well if the advisors are extremely honest or extremely dishonest. For example, BLADE discounts the ratings even if the advisor provides 70% honest reports. In the real world, advisors are not purely good or bad and could have various levels of honesty.

Vogiatzis et al. [11] proposed a probabilistic trust and reputation model that focuses on modeling service providers whose behavior is not static with time. Their model does not work well in the presence of biased advisors. Additionally, Vogiatzis's model and TRAVOS both assume that there has been a history of interactions between the agent (i.e, the reputation requester) and a service provider. Noorian et al. [5] categorize an advisor's "unfairness" behavior into two groups: intentional and unintentional. Their model, Prob-Cog, uses a two-layer filtering approach to detect and disqualify unfair advisors. Prob-Cog mainly targets and filters out advisors who are intentionally biased. Their model does not perform well when there is a large population of intentionally unfair advisors.

Zhang and Cohen [15] proposed a personalized approach to handle unfair ratings. They use private and public reputation information to evaluate the trustworthiness of advisors. They estimate the credibility of advisors using a time window to calculate the recency of ratings, and then estimate the trustworthiness of advisors based on the ratings. Their model does not interpret unfair ratings. As a result, when the proportion of unfair ratings increases, the trustworthiness of advisors decreases; this results in the system relying heavily on private reputation (i.e., agent's direct experiences). Yu and Singh [14] measure how much the advisor's rating deviates from the consumer's experience. Their model identifies accurate advisors, and discards deceptive advisors.

Another area of research is focused on sentiment analysis and review helpfulness. For example, Kim et. al. [3] propose a method for automatically determining the quality of reviews. They use regression to rank different sets of reviews on Amazon.com, based on their helpfulness. They do not customize the reviews based on a user's experiences or preferences. Also, since many products receive very few reviews, their approach is not helpful for such cases.

In contrast to these mentioned models, PRep uses and customizes reviews (or reports) even when they are biased. Without prior interactions with a service provider, a PRep agent can form a view about the service provider by requesting and interpreting the opinion of an advisor, if it has previously observed the advisor's behavior. This allows PRep agents to form a view about service providers that have very few reviews and ratings or for whom the majority of the reviews is biased. Other reputation models do not work effectively in such cases.

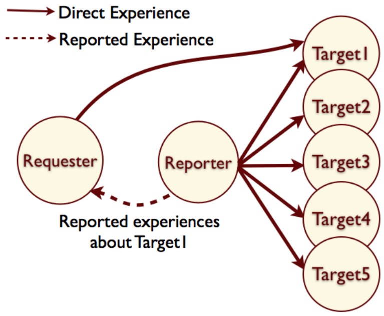


Figure 1: Basic scenario. Requester stands for Reputation Requester, Reporter stands for Reputation Reporter, and Targets are agents that Requester would like to know about.

3. THE PREP MODEL

In this section, we explain our reputation mechanism, PRep, which is based on probabilistic modeling and Bayesian learning. PRep has two main steps: learning the reporter's behavior (Section 3.3) and interpreting the later reports coming from that reporter for use in decision making (Section 3.4).

Figure 1 explains our model using a two-step scenario involving a reputation *Requester*, a reputation (review or opinion) *Reporter* (advisor), and several *Target*s (service providers). In this scenario, Requester is new to a society of agents, but Reporter has been in this society for some time and has had direct interactions with several agents (Target1, Target2, Target3, etc.). Requester first starts to interact with Target1 directly, then asks Reporter for some information about Target1. By comparing its own direct experience to the reported experience of Target1, Requester learns Reporter's reporting behavior. At this point, Requester can interpret acquired reports from Reporter about other agents (e.g., Target2) and can use this information to interact more effectively with those agents. Note that Target1 does not know that Requester is new and has requested reputation information from Reporter. This assumption prevents Target1 from deliberately misleading the Reporter in order to mislead the Requester. Also, Reporter does not know whether Requester has already interacted with Target1. The latter assumption prevents Reporter from deliberately misleading Requester about its reporting behavior.

Trust and reputation have generally been modeled using two sources: direct and reported experiences. PRep interprets reported experiences in its reputation model and uses a direct-experience trust model to evaluate the trustworthiness of agents. In this paper, we use HAPTIC [9] as the trust model. However, PRep is general and can be combined with other existing direct-experience trust models.

3.1 Direct-Experience Trust Model

Harsanyi Agents Pursuing Trust in Integrity and Competence (HAPTIC), a trust-based decision framework, is among the few existing models with a strong theoretical basis: HAPTIC is grounded in game theory and probabilistic modeling. It has been shown that HAPTIC agents are able to learn other agents' behaviors reliably using direct experiences. One shortcoming of HAPTIC is that it does not support reported experiences.

The HAPTIC model allows an agent to predict a partner's actions and use these predictions to decide whether or not to trust that partner. The key insight in HAPTIC is that it separately models trust using two components of *competence* and *integrity*. Competence is modeled as the probability that a given agent will be able to execute an action in a particular situation. Integrity is an agent's attitude towards honoring its commitments (or equivalently as the agent's belief in a discount factor), and is affected by the perceived probability of future interactions. This distinction is useful when an agent defects. It is important for the other agent to understand whether the defection was due to the incompetence of an honest agent, or was the result of cheating by a competent agent with low integrity. HAPTIC identifies a discrete set of player types, denoted by Θ, and maps each agent's competence and integrity θ to a value from this set. A HAPTIC agent observes the behavior of other agents and estimates their competence and integrity, then uses this data for decision making in future interactions with each agent.

HAPTIC has been applied to a modified two-player Iterated Prisoner's Dilemma (IPD), in which the payoff matrix in each round is scaled using a random multiplier. As a result, the payoffs differ from one round to the next. HAPTIC assumes that agents know the current round's multiplier before selecting their actions. With variable payoffs, a failure due to low competence can be distinguished from a failure that results from low integrity. An honest but incompetent agent defects randomly, irrespective of the payoff. By contrast, a cheating agent shows a pattern in its defections that is correlated with the expected payoffs. A HAPTIC agent computes expected payoffs (as defined in the classic Prisoner's Dilemma payoff matrix) and decides rationally whether to cooperate or defect. Equation 1 defines δ, a threshold for cooperation and defection. If the agent's integrity is greater than δ, it will cooperate; otherwise, it will defect. δ can be computed for each agent and each game using the current round's payoff multiplier *m*, the average payoff *M*, and the estimates of the payoffs of the four possible outcomes ($\widehat{P}$, $\widehat{S}$, $\widehat{R}$, and $\widehat{T}$).[1]

$$\delta = \frac{1}{\frac{M(\widehat{P}-\widehat{R})}{m(\widehat{R}-\widehat{T})}+1}. \tag{1}$$

A learning HAPTIC player considers the outcome of each round as either a Success (expected action) or a Failure (unexpected action), based on its hypothesis about that agent's type. Iterative games between two agents allow HAPTIC players to reduce the set of probable types being considered. The HAPTIC learning method uses observations of agent behavior to estimate the competence and integrity for each agent.

3.2 Types of Reporters

One of the dominant recognized reviewer behaviors (including eBay's Feedback Forum) is being positively or negatively biased. In the real world, some reviewers are realistic (and honest), truthfully providing the requested information, reviews, or rates. Others tend to hide people's defects because they are afraid of retaliation [7], they are hopeful of getting a good rate in return [1], or they gain personal or economic rewards or incentives by doing so. Still others may change the results with pessimism, because they are pessimistic people by nature, or because they want to ruin a competitor's reputation and discredit them. Note that reporting negatively about a service can be completely realistic and not pessimistic, if the service was actually bad.

To address the consequences of these behaviors in the real world, we model the behavior of reporters in PRep as being potentially biased. We define the reporters' behavior using three types: *realistic*, *optimistic*, and *pessimistic*, similar to Noorian et al.'s approach [5]. A realistic Reporter always reports truthful information, corresponding directly to the experiences that it has had in the past with other agents. A pessimistic Reporter underestimates other agents' behavior, and an optimistic Reporter overestimates other agents' behavior. The level of optimism (or pessimism) is modeled by an ordered pair, ω = (ω_{opt}, ω_{pess}), which may be based on the Reporter's innate characteristic or could depend on Reporter's incentives for honesty/dishonesty. Specifically, with probability ω_{opt}, Reporter will change some of the Defect actions of the target into Cooperates in its reports. Similarly, ω_{pess} defines the probability of changing Cooperate actions into Defects. For optimistic reporters, ω_{opt} represents the degree of optimism (probability of a $D \rightarrow C$ "flip"), and ω_{pess} is zero. Likewise, for pessimistic reporters, ω_{pess} is the degree of pessimism, and ω_{opt} is zero.

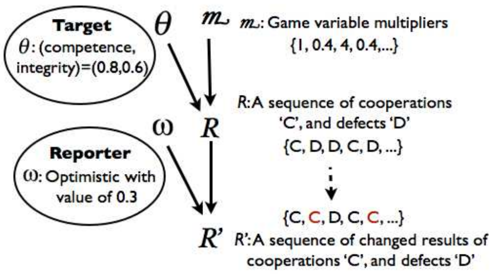

Target
θ: (competence, integrity)=(0.8,0.6)
θ
m
m: Game variable multipliers
{1, 0.4, 4, 0.4,...}
R:A sequence of cooperations 'C', and defects 'D'
{C, D, D, C, D, ...}
ω
R
Reporter
ω: Optimistic with value of 0.3
{C, C, D, C, C, ...}
R'
R':A sequence of changed results of cooperations 'C', and defects 'D'

Figure 2: Report generation from a game between Reporter and Target.

Figure 2 shows how a report is generated in an IPD environment, and how it will be changed by different reporters. We denote the actual result of the series of games between Reporter and Target as R. R is a sequence of Cooperate and Defect actions by Target in the series of games played with Reporter. The interactions and reporting process are as follows. Target makes its decisions based on its competence and integrity, θ, and the payoff multiplier *m* of each game, as modeled in HAPTIC. When Reporter wants to submit R to Requester, it will first change R to R' based on its type, ω, and then deliver R' to Requester. For example, if ω is 30% optimistic, then Reporter will change each Defect (in R) to a Cooperate (in R') with probability 0.3 (Figure 2).

In the real world, a Reporter could have various perceptions of interacting with different targets, based on its relationship with those targets, e.g., as a collaborator or competitor. Here, however, we assume that Reporter has the same perception of plays with different Targets, so its reporting behavior will be the same for various Targets. Since HAPTIC assumes that agents know the current multiplier of the round, we maintain this assumption here: all agents know the multipliers of the games. We intend to relax both of these assumptions in our future work.

3.3 Learn Reporter's Type

We now explain how Requester learns Reporter's type using Bayesian model averaging by comparing direct and reported experiences. Consider our basic scenario, shown in Figure 1, in which Requester and Reporter have played separately with Target1. Suppose that Requester asks Reporter for some information about Target1. We denote the actual results of the play between Reporter and Target1 by R, and between Requester and Target1 by D. Reporter changes the true results, R, based on its type, ω, to R' for reporting to Requester.

We define a set of discrete reporter types, Ω.[2] Each type $\omega_i \in \Omega$

[1] R, T, S, and P are the standard PD payoffs from the payoff matrix.

[2] Using a discrete set of possible agent types is simpler and less

is a pair of values (ω_{opt}, ω_{pess}). Realistic agents are modeled by ω = (0, 0). The probability of a type hypothesis ω_i is denoted by $P(\omega_i)$. Requester has also learned a probability distribution over the possible player types for Target1, which are denoted by θ_j. The probability of each player type is denoted by $P(\theta_j)$.

To find the probability of each type of Reporter, given the results R' and D, i.e., $P(\omega_i|R', D)$ for each Reporter type, ω_i, we use Bayesian model averaging over all possible Target1 types, θ_j:

$$P(\omega_i|R', D) = \sum_{\theta_j \in \Theta} P(\omega_i|R', D, \theta_j) \times P(\theta_j|R', D). \qquad (2)$$

The second term, $P(\theta_j|R', D)$, is the probability of Target1's type being θ_j, given R' and D. In this case, D, the direct experience, is more reliable than R', the reported experience. Therefore, PRep conditions θ_j only on D, and this term is simplified as $P(\theta_j|D)$, which is Requester's probability distribution of Target1's type, learned using the HAPTIC model.

The first term, $P(\omega_i|R', D, \theta_j)$, is the probability of a Reporter's type, given Target1's type θ_j, R', and D. Since ω_i is conditionally independent of the results of Requester and Target1's play (D) given θ_j and R', this term can be simplified to $P(\omega_i|R', \theta_j)$. Using Bayes's rule, we can rewrite this term as:

$$P(\omega_i|R', \theta_j) = \frac{P(R', \theta_j|\omega_i) \times P(\omega_i)}{P(R', \theta_j)}. \qquad (3)$$

We assume a uniform prior on the Reporter's type, so $P(\omega_i)$ is just the reciprocal of the number of defined types for Reporter ($P(\omega_i) = \frac{1}{|\Omega|}$). Also, $P(R', \theta_j)$ is a normalizing factor, so we only need to compute $P(R', \theta_j|\omega_i)$. Using the definition of conditional probability, this term can be rewritten as:

$$P(R', \theta_j|\omega_i) = P(R'|\theta_j, \omega_i) \times P(\theta_j|\omega_i). \qquad (4)$$

Since θ_j and ω_i are independent, the second term in Equation 4 is $P(\theta_j)$, a prior uniform distribution over the player types. The expected value of $P(R'|\theta_j, \omega_i)$ is defined by a weighted sum over all possible values of R:

$$E(P(R'|\theta_j, \omega_i)) = \sum_{R} P(R'|R, \theta_j, \omega_i) \times P(R|\theta_j, \omega_i). \qquad (5)$$

Since computing this full expectation is computationally very expensive, one can instead approximate $P(R'|\theta_j, \omega_i)$ using the maximum likelihood value for R. Denoting the most likely R as R^*, this maximum likelihood can be written and expanded as:

$$P(R'|R^*, \theta_j, \omega_i) = P(R'_C, R'_D|R^*, \theta_j, \omega_i), \qquad (6)$$

where R'_C are all the cooperates and R'_D are all the defects in the report. Since each round played is assumed to be independent of the others, the probabilities of the observed defects and cooperates in the report are independent of each other, yielding:

$$P(R'_C, R'_D|R^*, \theta_j, \omega_i) = P(R'_C|R^*, \theta_j, \omega_i) \times P(R'_D|R^*, \theta_j, \omega_i). \qquad (7)$$

Each term in Equation 7 represents a series of i.i.d. (independent and identically distributed) observations from a Bernoulli distribution, so a binomial distribution can be used to compute the overall probability of each reporter type. The first binomial is the probability of observing a certain number of optimistic flips (i.e., the case where the intention R^* of Target1 is Defect and the report of that round, R', is Cooperate). The second binomial likelihood is the probability of seeing the observed number of pessimistic flips in the report. (when the intention R^* is Cooperate, but is reported as a Defect in R'). The expected success rate for the first binomial is the number of $D \rightarrow C$ flips over total number of Cooperates in the results, R'_C, that would be expected from a reporter with type ω_i. Similarly, the expected success rate for the second binomial is the number of $C \rightarrow D$ flips over R'_D. Note that a success in this context is a "flip": that is, when Reporter changes a Cooperate to a Defect, or vice versa. We multiply these two binomial likelihoods to compute $P(R'|\omega_i, \theta_j)$ in Equation 7. By averaging over all possible Target1 types, Requester can calculate the probability of each type of Reporter (Equation 2).

In more complicated environments, the Requester may have multiple reports from the same Reporter. In this case, we first learn the Reporter's behavior in each set of reports, and then use a weighted averaging function over all possible Reporter types, i.e., for N reports. In fact, to estimate the credibility of the learned ω in each transaction, we use the length of each report, i.e., the number of rounds for which two agents interacted with each other in each run:

$$P(\omega_i|R'_1, D_1, .., R'_N, D_N) = \frac{\sum_{j=1}^{N} P(\omega_i|R'_j, D_j) \times length(R'_j)}{\sum_{k=1}^{N} length(R'_k)}, \qquad (8)$$

where $length(R'_j)$ is the number of interactions reported in R_j. Note that as the number of rounds increases, the statistics become more accurate, leading to better results (see Section 4).

3.4 Report Interpretation

In the previous subsection, Requester learned Reporter's type. In this section, the maximum likelihood of the possible Reporter types (i.e., $P(\omega_i|R', D)$) will be used to interpret the reported results for new Targets. We illustrate how agents can use this interpretation to learn the player types (competence and integrity) of other targets with whom they have not previously interacted.

After learning Reporter's type, Requester asks Reporter for information about Target2, and uses its learned knowledge of Reporter's type to interpret the reported results (which are denoted by R'_2). Without loss of generality, we explain how to interpret the reports when Reporter's type is optimistic. Recall that ω_{opt} represents the probability of optimistic flips in the report and ω_{pess} represents the probability of pessimistic flips in the report. Using Equation 9, an "interpret" function estimates the total number of Cooperates ($count_{R_{2C}}$ in the actual results R_2) using $count_{R'_{2C}}$, as the total number of reported Cooperates in the sequence R'_2, $length(R_2)$ as the number of rounds in the play, and ω_{opt}. The difference between $count_{R_{2C}}$ and $count_{R'_{2C}}$ is the number of Cooperates that should be changed back to Defects to produce more accurate results, and saving the result as R^*_2.

$$count_{R'_{2C}} = count_{R_{2C}} + \omega_{i\_opt} \times (length(R_2) - count_{R_{2C}}). \qquad (9)$$

Requester now plays back the new results, R^*_2—generating an action as it would do if it were actually playing with Target2—and uses HAPTIC to update $P(\theta_j)$ for each possible Target2 player type, $\theta_j = (C, I)$. This distribution will continue to be updated in the online learning process between Requester and Target2, when they start their direct interactions. This knowledge will increase Requester's overall and mean payoff.

4. EXPERIMENTS

In this section, we present our experimental results. We show the performance of the learning and report interpretation components of PRep. We also compare the overall performance of PRep, HAPTIC, and TRAVOS in terms of learning accuracy and payoffs.

computationally expensive than modeling agent types with a continuous variable. We experimented with a continuous version, and the results are very close to what we obtain with discrete sets.

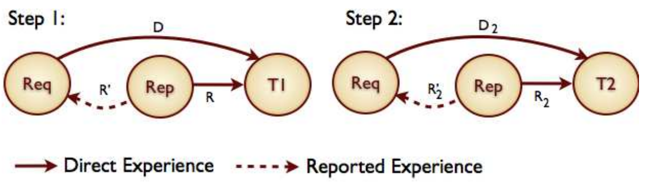


Figure 3: Step1 and Step2 of basic scenario. Req is Requester; Rep is Reporter; T1 & T2 are Targets.

As an overview, in the first two experiments, Exp1 and Exp2, we evaluate PRep's learning and interpretation components, respectively. In the third experiment, Exp3, we compare PRep with HAPTIC, and verify the results with a t-test. Finally, in Exp4, a TRAVOS Requester competes with a PRep Requester in finding Target1's behavior. We compare their mean error in finding Target1's behavior and the mean and cumulative game payoffs.

4.1 Simulation Parameters

Distribution of Reporter Types: In these experiments, the reporter type is chosen randomly using either a uniform distribution or a capped Gaussian distribution. These functions randomly generate numbers in the range (-0.7, 0.7), based on the type of distribution. A negative number represents a pessimistic reporter; a positive number is an optimistic reporter; and zero is realistic. We define the Gaussian distribution function with zero mean and a specified variance. Various demographics of realistic, pessimistic, and optimistic agents will be achieved by changing the variance of the Gaussian function.

PRep represents the set of possible reporters using a discrete set of types (ω_{opt}, ω_{pess}). Fifteen reporter types are considered by PRep: (0.1, 0), (0.2, 0)..(0.7,0) as optimistic reporter types; (0, 0.1), (0, 0.2),..(0,0.7) as pessimistic reporter types; and (0, 0) as a realistic reporter type. The uncertainty associated with the reporter's type is described by a multinomial probability distribution over these possible types. Learning of ω occurs by updating this probability distribution based on the observed behavior of that reporter after each reporting interaction.

Agent Strategies: Requester and Reporter are HAPTIC agents that have competence and integrity.[3] Targets are selected from classic strategies from the IPD literature in our experiments: ALLC, ALLD, TFT, and TFTT. An ALLC Target always cooperates in its play with any opponent. An ALLD Target always defects. A TFT (Tit-for-Tat) initially cooperates and then copies its counterpart's action from the previous round. A TFTT (Tit-for-Two-Tat) agent is forgiving and defects only if the opponent agent has defected twice in a row. We also use two variable-payoff strategies from Smith and desJardins [9]: DHP (Defect on High Payoff) and DMP (Defect on Medium Payoff). A DHP Target defects only on high-payoff games, and a DMP defects on medium and high payoffs, and cooperates on low payoffs.[4]Among these strategies, TFT and TFTT are the only ones who behave in reaction to their opponent's actions. The rest select their actions based on their type and regardless of their opponent's move.

We also introduce a noise factor for each of these strategic types, corresponding to HAPTIC's notion of competence. This factor, which is the probability of the actual action to be equal to the intended action, is selected from this set: {0.7, 0.8, 0.9, 1}.

4.2 Exp1: PRep Learning

In our first experiment, we show the performance of PRep's learning component for different reporter types. We compare the given Reporter type distribution with the learned distribution and measure the accuracy of the learned Reporter types.

Design: We evaluate PRep in two steps, shown in Figure 3 (which follows our basic scenario presented in Figure 1). In the first step, PRep learns ω; in the second step, it uses the learned ω to interpret the reports in its successive plays. In step one, Requester and Reporter each play 20 rounds with Target1. Then, Requester asks Reporter about its experience with Target1. Reporter converts the actual results, R, to R' based on its type ω, and passes the report, R', to Requester. Requester then learns the Reporter's type, ω, given R' and R (using the approach described in Section 3.3). In step two, Reporter plays 20 rounds with Target2 (results = R_2). Then, Requester asks Reporter about Target2. Reporter converts the actual results R_2 to R'_2 based on its type ω, and passes the results to Requester. Requester interprets R'_2 based on the learned ω, and generates R^*_2.[5] Requester plays back R^*_2 and learns Target2's competence and integrity, denoted by (C, I). Finally, Requester plays for 20 rounds with Target2, starting with its learned values for Target2's (C, I).

In Exp1, 100 Reporter types, ω, are selected randomly from a uniform or Gaussian distribution. Requester and Reporter player types (Competence, and Integrity) values are (1, 0.9). Target1 and Target2 types are selected randomly from a set of 16 strategic types: namely, the cross products of 4 player types (ALLC, ALLD, DHP, and DMP) and 4 competence values (0.7, 0.8, 0.9, 1).

As a performance metric, we use the mean error, which is the difference between the identified Reporter type, ω, and the correct type. All results are averaged over 100 runs. [6]

Results: Figure 4(a) shows the distribution of true reporter types and most likely learned types in 100 runs of the experiment over 100 reporter types, when the true reporter types are selected using a uniform distribution. PRep is able to identify the uniform distribution, since the values are almost equally spread over the optimistic and pessimistic ranges, except for the realistic type (which will be explained next). The mean error for this experiment is 0.14. Part of this error arises from using discrete types in the learning process: the discrete steps are 0.1, so inherently an error up to 0.05 will be introduced during learning (0.025 on average).

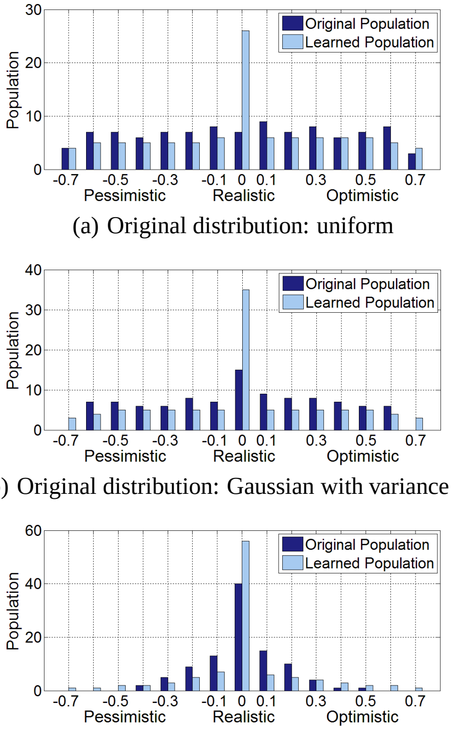
Another source of error is the population of learned realistic reporters (ω = 0), which is much higher (about 28) than the true number of realistic reporters value (100/15 or around 7). The explanation for this disparity is that optimistic reporters cannot be identified when they are reporting about ALLC players. An ALLC player always cooperates, so an optimistic reporter makes no changes in the report, and PRep detects such reporters as realistic. This problem can be solved when a PRep agent has multiple encounters with the same reporter (see Section 4.5). The same is true for pessimistic reporters when reporting about ALLD players. The population of ALLC players is 25, and roughly half those will face an optimistic reporter, which is 12 in the population. Similarly, another 12 false positives are generated from the ALLD players. Therefore, the population of realistic reporters will be estimated as 24 more than the true number. Since these misidentified realistic Reporters have a true value between 0.1 and 0.7, the average error for each of these

[3] As in Smith and desJardins, competence of agents are selected from {0.7, 0.8, 0.9, 1}; and integrity is a number from 0 to 1.

[4] Multipliers of the rounds are selected from {0.4,1,4}. A DHP defects on rounds with m=4 and DMP defects on m=1 and 0.4 [9].

[5] R^*_2 is Requester's estimation of what actually happened between Reporter and Target2, as R_2 is not available to Requester.

[6] Note that a "run" is different than a "round." A "round" is a single play between two agents in PD game, with single Cooperate or Defect as outcome. A "run" is a combination of several "rounds" in games between the agents in a scenario.



(a) Original distribution: uniform

(b) Original distribution: Gaussian with variance 0.3

(c) Original distribution: Gaussian with variance 0.1

Figure 4: Exp1; The probability associated with Reporter's true reporting type.

24 Reporters will be 0.4. This will cause an additional 0.096 (i.e., 0.4×24/100) error, making the estimated overall error to be 0.121 (0.025 + 0.096), which is very close to the actual error.

Figures 4(b) and 4(c) show the distribution of true agent types and most likely learned types over 100 reporter types, ω, selected from a Gaussian distribution, with variances of 0.3 (15% realistic) and 0.1 (41% realistic reporters), respectively. PRep is able to identify different distributions of reporters and the learned population is close to the original population for both large and small variances in the Gaussian function. The mean error for variance 0.3 is 0.11 and for variance 0.1 is 0.077. As the number of realistic reporters increases in the population, the mean error decreases; this occurs in part partially because fewer ALLC and ALLD targets will face optimistic or pessimistic reporters, respectively.

4.3 Exp2: PRep Interpretation

In the second experiment, our goal is to show the importance of correct interpretations when a reporter is biased. We design an experiment with fixed values (as a snapshot of Exp1), average it over 100 runs, and focus on finding the correct Reporter's type, ω and the Target's type, (C, I), after report interpretation and the resulting cumulative payoff.

Design: We follow the scenario shown in Figure 3. In the first step, Requester and Reporter play 30 rounds with Target1. In the second step, Requester and Reporter play 20 rounds with Target2.

In this experiment, we use HAPTIC as a baseline. Also, to show the negative effect of not re-interpreting reports, we define another baseline, PRep-NoInterp. This baseline uses PRep model without the interpretation component. A third baseline, PRep-Perfect, shows the upper limit benefit of reported experiences when the reporter is realistic and there are no flips in the report.

In Exp2, Reporter's true type is optimistic 0.4. Requester and Reporter player types values are fixed at (1, 0.9). Target1's (C, I) is: (1, 0.6), and Target2's true value is (0.7, 0.6).

Our performance metrics are the accuracy of the learned Reporter's ω and Target2's player types (by looking at the probability assigned to the true player types, i.e., (C, I)) and the cumulative payoff. The results are averaged over 100 runs.

Results: Figure 5(a) shows the results of learning Reporter's ω in Exp2, averaged over 100 runs, where ω is optimistic 0.4. This graph shows that Player1 was able to identify Reporter's type as having an optimistic behavior. The probability of the levels of optimism is spread over different values; the maximum likelihood of these values, is optimistic 0.4, with probability 0.22. This result illustrates the correctness of PRep's learning component.

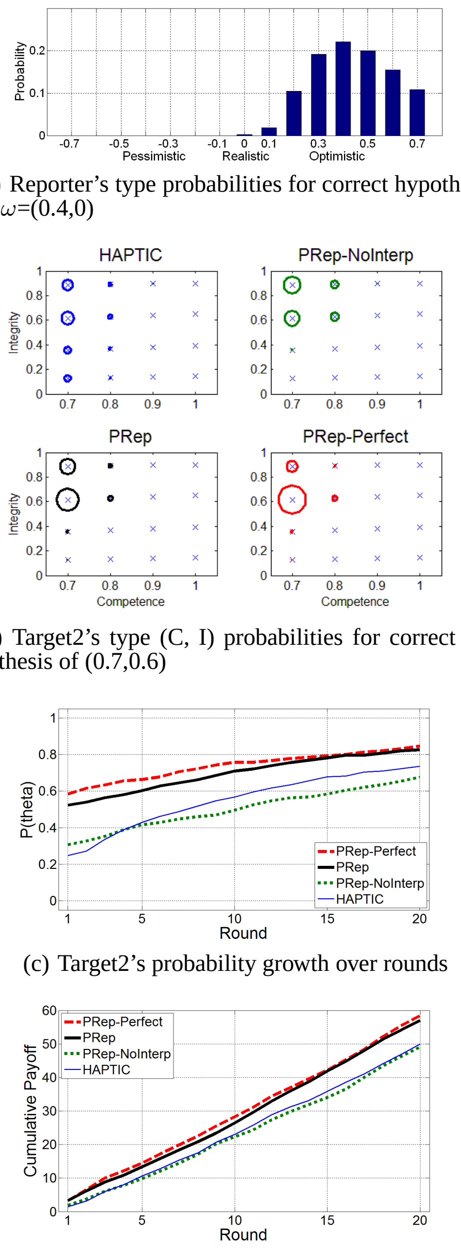
Figure 5(b) displays the results of learning Target2's (C, I). The possible hypotheses for Player4 are shown by small cross signs; the correct hypothesis is (0.7, 0.6), which is the true value of Target2 type. The circles' sizes represent the learned probability of each hypothesis for Target2. The top left graph shows the results for HAPTIC. In this case, Requester uses only direct experiences. After 20 rounds of play, the hypothesis probabilities are spread among four values: (0.7, 0.9), (0.7, 0.6), (0.7, 0.35), and (0.7, 0.1), which means that Requester is getting close but has not yet correctly identified Target2's true type. The PRep-NoInterp graph shows that using the non-interpreted reports still yields a moderate probability of finding the correct hypothesis. The results for PRep are shown in the bottom left graph, where the highest probability is assigned to (0.7, 0.6). This is the correct hypothesis; therefore, Requester can achieve higher payoffs with this learned model than using direct experience alone. If Reporter was a realistic reporter instead of being 40% optimistic in Exp2, Requester would have been able to identify Target2's actual (C, I) with a higher probability, as shown in PRep-Perfect graph in Figure 5(b).

Another interesting view of the learning process is how the learned probabilities changes over a series of rounds for Target2's true type. As seen in Figure 5(c), PRep starts high (near 0.56) from the beginning, while HAPTIC's probability of the true type remains at a lower level and needs several more rounds to increase. The main reason for this behavior is that PRep has learned Target2's type using reported experiences that it has received from Reporter.

The corresponding payoffs resulting from the four approaches are shown in Figure 5(d). As expected, PRep-Perfect has the highest payoff; PRep (that interprets biased reports) ranks second and yields payoffs close to PRep-Perfect. HAPTIC places third; PRep-NoInterp is in the fourth place and behaves very similarly to HAPTIC. Since the reporter in this experiment alters Defects in the results with a 40% probability, using reports without interpretations will result in a performance close to HAPTIC, which is hindered by its belief in the incorrect reports.

4.4 Exp3: HAPTIC Vs. PRep

To verify the effectiveness of PRep over different player and reporter types, we performed Exp3, repeating a game for 100 times. In each run, we use the scenario in Figure 3. Requester's type is fixed at (1, 0.9), and the reporters' types are selected based on a Gaussian distribution with 0.3 variance (15% realistic reporters) centered on zero. The Target1 and Target2 types are selected randomly among 16 strategic types: the cross product of four strategic types (ALLC, ALLD, DHP, and DMP) with 4 competence values (0.7, 0.8, 0.9, 1). The mean payoffs for HAPTIC, PRep, and PRep-Perfect in this experiment are 1.89, 2.17, and 2.18, respectively. PRep (with biased reporters) achieves 14.8% improvement over HAPTIC, where the upper limit is 15.3% achieved by PRep-Perfect. A t-test confirms that the mean per-round payoffs of HAP-



(a) Reporter's type probabilities for correct hypothesis of ω=(0.4,0)

(b) Target2's type (C, I) probabilities for correct hypothesis of (0.7,0.6)

(c) Target2's probability growth over rounds

(d) Cumulative payoffs

Figure 5: Exp2; A 40% optimistic Reporter and Target2 type actual values (C= 0.7, I = 0.6).

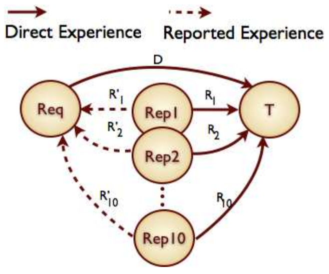


Figure 6: Scenario for TRAVOS and PRep. Req is Requester; R1, R2,..R10 are Reporters; and T is the Target.

TIC and PRep are different; with 99.9% confidence, the difference is between 0.274 and 0.276.

4.5 Exp4: TRAVOS Vs. PRep

In Exp4, we compare the performance of TRAVOS [10] and PRep in a noisy environment with biased and unbiased reporters. We measure the accuracy of the learned Target types, and the resulting mean and cumulative payoffs for both a PRep Requester and a TRAVOS Requester.

TRAVOS: This model uses probabilistic modeling based on a beta distribution. TRAVOS outperforms many other trust and reputation models, including probabilistic models such as BRS [13]. TRAVOS uses the number of satisfactory and unsatisfactory interactions with the sellers as ratings, and uses a weight function to combine these ratings. Agents calculate rating weights by comparing the relevance of each rating to their own opinions.

TRAVOS models the trustworthiness of each agent by a fulfillment factor, which is equivalent to "competence" in PRep. However, TRAVOS does not model the integrity of an agent. In order to compare PRep and TRAVOS, we settle this difference by providing the integrity of an agent as an input to TRAVOS, whereas PRep is searching in a two-dimensional space for competence and integrity. Note that this gives an advantage to TRAVOS.

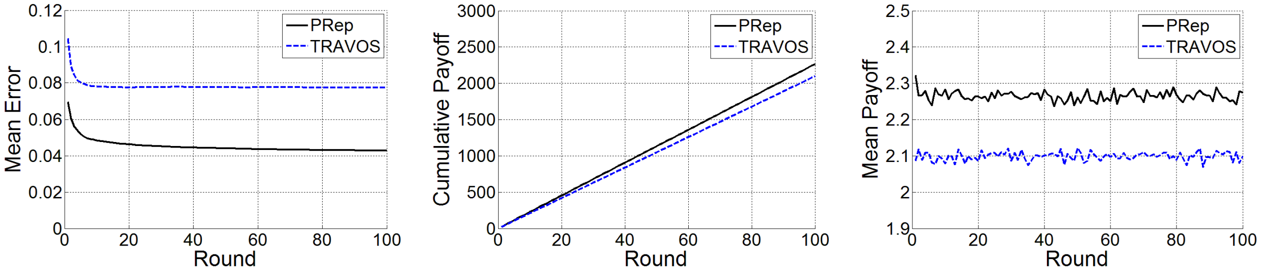
Design: To be able to compare PRep and TRAVOS in both modeling and assumptions, Exp4 uses another IPD-based test framework. TRAVOS assumes previous transactions between Requester and Target, so we design this experiment with this assumption. Also, we have several Reporters (each with different behavior) in this experiment reporting about one Target. Therefore, the Requester interprets different reporters' reports about one Target.

The scenario for this experiment is shown in Figure 6. Requester plays with Target for 10 rounds. Ten Reporters play 10 rounds with Target. Each Reporter changes the outcome of its play based on its type and then reports the changed results to Requester, who updates its belief about that specific Reporter. We repeat the same process 100 times; in each run, a Reporter's type, ω, is learned. In PRep, this value will be averaged over the so far learned ω (as seen in Equation 8) and later will be used in interpreting reports.

In this experiment, Requesters use either TRAVOS or PRep for modeling their trust and reputation; target types are selected randomly from the cross product of six strategic types (ALLC, ALLD, DHP, DMP, TFT and TFTT) with 4 competence values (0.7, 0.8, 0.9, 1). Requester and Reporter's competences and integrities are fixed at (0.8, 0.9). The population of Reporters consists of realistic and biased reporters (pessimistic/optimistic up to 0.7 and realistic), selected from a Gaussian distribution with 0.1 variance (41% realistic reporters) centered on zero.

We compare the accuracy of Target player types (competence in this experiment) learned by TRAVOS and PRep. As a performance metric, we use the mean error, which is the difference between the identified type and the correct type. Also, we compare PRep and TRAVOS in terms of the mean and cumulative payoff. All results are averaged over 100 runs.

Results: Despite the fact that we have provided TRAVOS with the correct integrity, as we can see in Figure 7(a), PRep outperforms TRAVOS in identifying the Target's type (competence). This error for TRAVOS has converged to 0.078 and for PRep to 0.043 (a 45% improvement over TRAVOS). The reason is that TRAVOS heavily discounts the biased reports, while PRep interprets and uses that data to learn more about the behavior of the Target. As a result of correctly identifying the behavior of the Reporter, the cumulative payoff is increased from 2085 to 2264 (Figure 7(b)) and the average payoff per round is increased from 2.09 to 2.26 (a 9% im-



(a) (b) (c)

Figure 7: Exp4; TRAVOS vs. PRep; (a) Mean error in identifying true Target's competence, (b) Cumulative, and (c) Mean payoffs for Requester in its play with Target.

provement), as shown in Figure 7(c). The results passed the t-test, which verifies the mean values of TRAVOS and PRep are different; with 99.9% confidence, the mean payoff difference is between 0.16 and 0.17.

We repeated this experiment with various number of rounds of direct experiences (i.e., "D" in Figure 6). The results show that TRAVOS is performing as well as PRep when the number of direct experiences is high. Figure 8 shows that the mean error of both models converges to the same value if we increase the number of direct interactions up to 30. This means that TRAVOS is heavily relying on direct experiences, and PRep is performing better when there are only a few direct interactions available. Additionally, it shows that mean error decreases for both TRAVOS and PRep when the number of realistic reporters increases in the population.

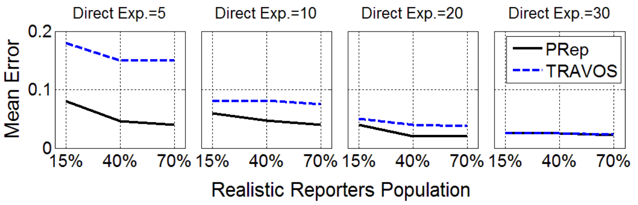


Figure 8: Exp4; Performance of TRAVOS vs. PRep using a variable number of direct experiences.

5. CONCLUSIONS AND FUTURE WORK

We presented PRep, a reputation mechanism that is capable of re-interpreting and adjusting reported experiences by learning the reporters' behavior. PRep works well in regular and noisy environments, even with the presence of a large population of biased reporters, and when there are only a few direct interactions available. Our results show that a PRep agent identifies agents' reporting behavior correctly; therefore, it recognizes other agents' trustworthiness more rapidly and accurately than a HAPTIC or TRAVOS agent, resulting in better decision making. For example, with 10 direct interactions, PRep's mean error for predicting an agent's behavior is 45% lower than that of TRAVOS, due to PRep's ability to correctly interpret the reports. As a result, the average payoff improves by 9%.

An interesting direction for future work would be to further evaluate this model in a risky and non-deterministic environment, such as a marketplace application. Also, we plan to explore the use of context-dependent Reporter types that can cause agents to behave differently in various situations (e.g., when reporting to a competitor versus a collaborator). We will also investigate multidimensional trust models that can be applied when a Reporter can have varying degrees of trust for different aspects of a Target's behavior (e.g., quality and price in a supply chain management context).

6. REFERENCES

[1] M. Chen and J. Singh. Computing and using reputations for internet ratings. In *Proceedings of the 3rd ACM Conference on Electronic Commerce*, pages 154–162, 2001.

[2] A. Josang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, 2007.

[3] S. Kim, P. Pantel, T. Chklovski, and M. Pennacchiotti. Automatically assessing review helpfulness. In *EMNLP-06*, pages 423–430, 2006.

[4] L. Mui, M. Mohtashemi, and A. Halberstadt. Notions of reputation in multi-agent systems: A review. In *AAMAS-02*, pages 280–287, Bologna, Italy, July 2002.

[5] Z. Noorian, S. Marsh, and M. Fleming. Multi-layer cognitive filtering by behavioral modeling. In *AAMAS 2011*, pages 2–6, Taipei, Taiwan, 2011.

[6] K. Regan, P. Poupart, and R. Cohen. Bayesian reputation modeling in e-marketplaces sensitive to subjectivity, deception and change. In *AAAI-06*, volume 21, page 1206. AAAI Press, 2006.

[7] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman. Reputation systems. *Communications of the ACM*, 43(12):45–48, 2000.

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

[9] M. Smith and M. desJardins. Learning to trust in the competence and commitment of agents. *Journal of AAMAS*, 18(1):36–82, February 2009.

[10] W. Teacy, J. Patel, N. Jennings, and M. Luck. Coping with inaccurate reputation sources: Experimental analysis of a probabilistic trust model. In *AAMAS-05*, pages 25–29, 2005.

[11] G. Vogiatzis, I. MacGillivray, and M. Chli. A probabilistic model for trust and reputation. In *AAMAS-10*, pages 225–232, 2010.

[12] Y. Wang, C. Hang, and M. Singh. A probabilistic approach for maintaining trust based on evidence. *JAIR*, 40:221–267, 2011.

[13] A. Whitby, A. Jøsang, and J. Indulska. Filtering out unfair ratings in Bayesian reputation systems. In *Proceedings of the 7th Int. Workshop on Trust in Agent Societies*, 2004.

[14] B. Yu and M. Singh. Detecting deception in reputation management. In *AAMAS-03*, pages 73–80, 2003.

[15] J. Zhang and R. Cohen. Evaluating the trustworthiness of advice about seller agents in e-marketplaces: A personalized approach. *Electronic Commerce Research and Applications*, 7(3):330–340, 2008.