Bootstrapping trust and stereotypes with tags

Caroline E. Player
Department of Computer Science
University of Warwick, CV4 7AL
c.e.player@warwick.ac.uk

Nathan Griffiths
Department of Computer Science
University of Warwick, CV4 7AL
nathan.griffiths@warwick.ac.uk

Abstract

In real-world environments, cooperation often emerges amongst agents who are observably similar. Estimating the expected behaviour of another agent is a challenging problem, particularly for new agents who have little or no experience of others. In this paper, we show how observable features can be used to find similar, and hence cooperative, partners.

Our contribution extends trust and stereotype approaches, to include comparisons and learning of observable features, called *tags*. In environments where no reciprocity exists (or where there have been insufficient interactions for reciprocity to take effect) tags have been used to encourage cooperation. The only information available to an agent early in its life is knowledge of its own tags and behaviour. We assume that agents who are observably similar will be behaviourally similar too. Agents use reinforcement learning to take advantage of as much available information as possible, until sufficient experience has been gathered for more established trust and stereotype models to be built.

Our results show that using tags improves agents' rewards in the early stages of their lifetime when used prior to established stereotype and trust algorithms. We demonstrate that tags are successful in supporting cooperation, even when agent behaviour is independent of the partner, because the approach correctly identifies similar agents. Good agents are able to select partners who will act as they do, while bad agents avoid those who are observably similar.

1 Introduction

Trust in multi-agent systems (MAS) encourages cooperation amongst agents without the need for a central authority or rules, which can be expensive and impossible to impose. We adopt Gambetta's definition of trust, namely, the amount of trust you have in another agent is the utility you expect to receive from that agent when that behaviour impacts on your future [Gam00]. This is important for agents choosing partners that will maximise their expected reward. However a situation faced by new agents is having insufficient information with which to calculate trust.

In: J. Zhang, M. Sensoy, and R. Falcone (eds.): Proceedings of the 19th International Workshop on Trust in Agent Societies, Sao Paulo, Brazil on May 8-9, 2017, published at http://ceur-ws.org

Calculating trust and reputation requires a significant number of experiences with the agent. Furthermore, agents can get stuck in local maxima when choosing their partners because they continue to interact with the first agent(s) they find who are better than average.

One method to improve estimates of a new agent's behaviour, is to assume that it will act the same as agents who are observably similar. Known as *stereotype-based trust*, such models typically use machine learning techniques to learn these estimates, but they require large amounts of representative data [BNS10, LDRL09]. Stereotypes help existing agents evaluate new agents in the system despite not having direct experiences with them. They do this by using the experiences they have accumulated in their lifetime to learn which observable feature values of agents they prefer when selecting a new partner. However, stereotypes do not address the problem for a new agent to choose an interaction partner, as they have not yet collected the experiences with which to formulate preferences.

Choosing a partner in the early stages of an agent's lifetime needs to strike a balance between exploration and reward. We use *tags*, a concept from evolutionary biology [Daw76, Ham64], which have been found to encourage cooperation in MAS [Axe84, AD$^+$88, AHG04, HO05, YCS15]. Tags are notionally equivalent to the features used in stereotype models. Agents know their own tags and how they themselves behave, and so they can make assumptions about another agent's behaviour if they have similar tags. As with stereotype models, we assume that there is a correlation between tags and behaviour, and in line with evolutionary studies we assume that this behaviour is more altruistic amongst similar agents. However, existing tag-based cooperation techniques have not considered identifying features that do not correlate with behaviour or identifying agents who behave well despite having different tags.

In this paper we propose a method for using tags to bootstrap stereotypes and trust, with the aim of improving new agents' rewards. Such agents have no prior experience and so cannot assess others' trustworthiness, and do not know which tags are indicative of good behaviour or are irrelevant. The tag-based system is complemented by trust and stereotype assessments once sufficient experiences are available, allowing agents to obtain accurate assessments later in their lifetime. Once they start to gain experience, agents can use reinforcement learning to improve their estimates of trust.

The remainder of this paper is structured as follows. In Section 2 a discussion of existing techniques is presented. Section 3 describes the context of this work and how it translates to MAS. The specifics of using tags to bootstrap stereotypes and trust is presented in Section 4. The results and analysis are given in Section 5, and Section 6 concludes the paper.

2 Related Work

Trust and reputation assess the belief and uncertainty in agents' expected behaviour. ReGreT is a successful algorithm which aggregates the values of experiences (where experiences are classified as positive or negative), and weights them by how old each experience is [SS01]. This multi-faceted model assumes aspects ranging from social connections, to price and quality affect an agent's behaviour. Trust algorithms such as ReGreT can be successful in a specific environment though lack the generality required to be applied in new contexts.

Reputation models, which often work in conjunction with trust, include information from third parties as well as direct personal experiences when assessing expected behaviour of other agents. FIRE is a reputation model that extends ReGreT to tackle the cold-start problem in open MAS by using reputation information from more established agents [HJ04]. In this paper we demonstrate a proof of concept by bootstrapping the Beta Reputation System (BRS) [JI02], since it is simple and is the base of other more complex methods [BNS10, BNS13, TLRJ12, TPJL06]. The contribution of this paper is the bootstrap model, and an alternative reputation model could be used instead of BRS.

Stereotypes are a tool for existing agents of a system to assess others where no direct or indirect experiences are available. Stereotype models use past interactions with all agents, to discover whether their features correlate with good or bad behaviour. A new agent can then be classified by its features. Different stereotype learning techniques have been successful in different circumstances, such as clustering [Wag10], or finding patterns in a semantic web graph of the features [ŞYN16]. Bootstrapping trust with stereotypes to improve an agent's ability to classify a new potential partner, while allowing a trust algorithm to take precedence after interaction experiences have become available, has been shown to improve an agent's ability to find successful partners [ŞYN16, BNS10]. We extend the approach of Burnett *et al.* because of its use of the BRS reputation model and simple approach, where we can replace early estimations of behaviour with our tag-based model before the agent has collected the necessary amount of information to build a stereotype model.

Table 1: Tag types and their distance equations

Tag Type	Distance Equation, $dist(\overrightarrow{\tau_t^i}, \overrightarrow{\tau_t^j})$	Example
binary	Hamming Distance	$Hamming(1001, 1100) = \frac{2}{4}$
⟨latitude, longitude⟩	Euclidean Distance	$\sqrt{(long_1 - long_2)^2 + (lat_1 - lat_2)^2}$
ordered category	extent of difference	{very good, good, bad, very bad}. $dist$(very good, good) $= \frac{1}{4}$
category	different or not	{England, France, Germany, Italy}. $dist$(Germany, England) = 1 $dist$(England, England = 0)
nominal value	difference	$val_1 - val_2$

In evolutionary biology, animals are found to use observable features, called *tags*, to determine relatedness and therefore selective altruism [Ham64]. The *green beard* example illustrates this concept: those with a green beard, which can be formalised to a tag holding a value, are more likely to cooperate amongst themselves [Daw76]. Existing tag-based work typically assigns a tag and threshold to each agent, and an agent will choose a cooperative strategy if the difference between their two tags is within the threshold, thus enabling cooperation to emerge [HE03, RCA01]. These studies pursue social rationality, a phenomenon where agents optionally choose strategies that earn them less immediate reward but improve the group utility. This differs from trust and reputation which aim to accurately evaluate another agent's behaviour. However, we apply the assumption that an agent's behaviour is influenced by their partner's tags. Tags are similar to features, they are used to describe characteristics other than physical appearances in agents, such as location, points on a driving license, or criminal convictions. We use the terms tags and features interchangeably to mean any descriptor of an agent.

3 Agent Interaction Environment

The tag-based model introduced in this paper is depicted and evaluated in the context of ride sharing, although its generality makes it applicable to an online marketplace. The drivers and passengers are represented as a set of agents, $\mathcal{A}$.

3.1 Agents and Behaviours

An agent, $a_i \in \mathcal{A}$, may represent a driver or a passenger. The set of *trustors* are a subset of the agents who use the model to choose a partner to interact with. The other agents are *trustees*, who are being evaluated by the trustors, all connected in a complete bipartite graph. In our evaluation a passenger is a trustor, a_i, who chooses a driver, a trustee a_j.

Agent $a_i \in \mathcal{A}$, can be described by their *tags*, which are their observable features, denoted as the vector $\overrightarrow{\tau}_i$. Each tag corresponds to a characteristic, for example their location, and will be of a type listed in Table 1. The index of a tag corresponds to the same tag in another agent, making tag vectors comparable.

Trustors receive some utility/outcome as a result of interacting with a trustee. The utilities a trustee offers over multiple interactions are drawn from a Gaussian distribution and are not uniform, since environmental factors may cause variances in their capabilities at any time. We evaluate our work with two different types of behaviour. First, agents can have *simple behaviours* of the form used by Burnett *et al.* [BNS10]. Second, based on biases which have been found to occur in real world societies [Daw76, Ham64], agents using *biased behaviours* tend to act more cooperatively towards agents who are observably similar to them.

With simple behaviours, the utility that trustor a_i receives from trustee a_j's behaviour comes from a Gaussian distribution defined by mean, μ_j, and standard deviation, θ_j. An agent is assigned a profile which is associated with values for a mean and standard deviation then adopted by the agent, as well as their tags. Relevant features are those in the tag vector $\overrightarrow{\tau}$ that have the same value v for all agents of that profile, enforcing the correlation with that profile's behaviour. There are N_{nf} noisy features assigned randomly to every agent, for example not all features of a driver are an indication of the service they provide. Table 2 shows each profile's

Table 2: Profiles with non-binary tags, based on [BNS10]

Profile	μ	θ	$\overrightarrow{\tau}_0$, binary	$\overrightarrow{\tau}_1$, categorical	$\overrightarrow{\tau}_2$, location	$\overrightarrow{\tau}_3$, nominal	$\overrightarrow{\tau}_4$, ordered category	$\overrightarrow{\tau}_{5,..,5+N_{nf}}$
p_1	0.9	0.05	1	v_1	v_2	v_3	v_4	*rand*
p_2	0.6	0.15	0	0	v_2	v_3	0	*rand*
p_3	0.4	0.15	0	v_1	C	v_3	0	*rand*
p_4	0.3	0.1	0	0	C	0	0	*rand*
p_5	0.0	1	0	0	C	v_3	v_4	*rand*

behaviour distribution parameters and tags. In the case of simple behaviours, these parameters remain static in an interaction, regardless of the trustor they are interacting with.

Biased behaviours replace the static value of μ for each profile by the output of a function, $f(sim_{ij}) \rightarrow \mu_{ij}$. This new mean accounts for how trustee a_j specifically treats a_i. Similarity between two agents is the mean difference in value of every observable feature, as defined in Equation 1, and includes irrelevant tags as agents do not know which are relevant and which are not. The distance between the tth tag of each agent depends what type it is. Each tag type's distance calculation $dist(\overrightarrow{\tau_t^i}, \overrightarrow{\tau_t^j})$, can be seen in Table 1.

$$sim_{ij} = 1 - \frac{\sum_t^{|\overrightarrow{\tau}|} dist(\overrightarrow{\tau_t^i}, \overrightarrow{\tau_t^j})}{|\overrightarrow{\tau}|} \tag{1}$$

The profiles' behavioural functions are given in Table 3 (the standard deviation and tag vectors remain the same as Table 2). These functions are either sigmoid or exponential functions, transformed to create outputs in the range $[0, 1]$. Agents improve their behaviour if they are interacting with an agent they believe is observably similar to them, and the different profiles do this to more extreme extents as visualised in Figure 1. An agent of profile p_1 always has a high mean and this might denote a well established, highly reputable driver who is keen to perform well despite differences in similarity. Alternatively, p_3 agents only perform well when they are very similar to their partner, which might represent an agent who is trained in some very specific way thus only performing well in that circumstance.

Table 3: Behaviour functions for biased behaviours

Profile	Function
p_1	$f(sim_{ij}) = 0.8 + 0.2 \times (\frac{6(sim_{ij}-0.5)}{\sqrt{1+6(sim_{ij}-0.5)^2}})$
p_2	$f(sim_{ij}) = 0.5 + 0.2 \times (\frac{3(sim_{ij}-0.5)}{\sqrt{1+3(sim_{ij}-0.5)^2}})$
p_3	$f(sim_{ij}) = 0.5 + 0.4 \times (\frac{8(sim_{ij}-0.85)}{\sqrt{1+8(sim_{ij}-0.85)^2}})$
p_4	$f(sim_{ij}) = 0.2 + (sim_{ij} \times 0.7)^3$
p_5	$f(sim_{ij}) = 0.5$

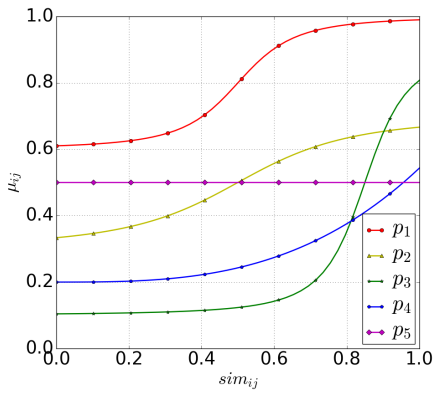


Figure 1: Biased behaviour functions.

3.2 Partner Selection

Trustors have the ability to choose the trustee they believe is the best, but they can get stuck in a local maximum when they find a set of agents who behave better than average and are ignorant to the existence of available, better agents. Therefore, we use simulated annealing to explore agents that would otherwise not be selected. The probability, p, of choosing a random agent a_j instead of the best available agent is dependent on the difference in their expected behaviour, ΔEB, and the current temperature T.

$$p = e^{\frac{\Delta EB}{T}} \tag{2}$$

The temperature should decrease as agent a_i increases their number of experiences n_i relative to their lifetime. A trustor tr can leave the system with probability P_{tr}, so its average lifetime is $\frac{1}{P_{tr}}$. The number of experiences they have as a proportion of their lifetime is therefore: $\frac{1}{P_{tr} \times n_i}$. We scale this temperature, by a coefficient ι, such that most their exploration is done within some proportion of their life.

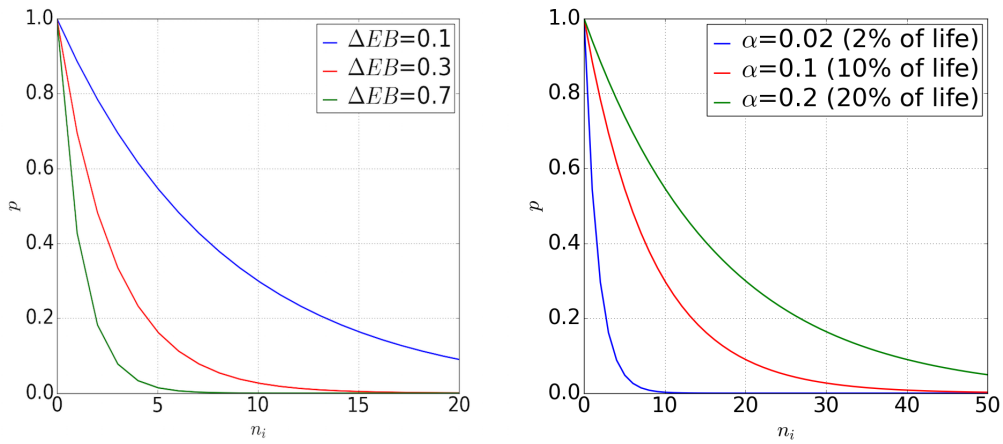
$$T = \frac{1}{n_i \times P_{tr}} \times \iota \tag{3}$$

Specifically, the coefficient, ι, scales the decay to achieve an exploration probability of higher than 0.3 in α percent of the agent's lifetime, if it is interacting with an agent of average difference in expected behaviour from the best known available agent. The average difference in expected behaviour, ΔEB, between a random agent a_j and the best known agent a_{best} is $trust(a_j) - trust(a_{best})$, and as $trust(a_j) < trust(a_{best})$, $\Delta EB = -0.5$. The value of 0.3 to probabilistically choose a random agent is an upper bound and other non-zero values could be used, with the magnitude determining the extent of initial exploration. Solving for ι, we have:

$$\iota = \frac{ln(0.3)}{-0.5 * \alpha} \tag{4}$$

Equation 5 gives the overall probability of selecting a random agent instead of the best. Figure 2a visualises this probability when faced with trustees of varying differences of expected behaviour between them and the best known agent. While Figure 2b shows how exploration decays more slowly as α increases, because we want the agent exploring for higher proportions of their lives.

$$p = e^{\frac{\Delta EB \times P_{tr} \times n_i \times ln(0.3)}{-0.5 \times \alpha}} \tag{5}$$



(a) Varying ΔEB ($alpha = 0.02$ and $P_{tr} = 0.01$) (b) Varying α ($\Delta EB = 0.5$ and $P_{tr} = 0.01$)

Figure 2: Average trust in profiles, agents use simple behaviours, TSB

3.3 Interactions and Histories

In an interaction between a_i and a_j, the utility u_i received by a_i is a real number drawn from a_j's behaviour distribution. Agent a_i records u_i as either a good or bad experience depending if u_i is above or below 0.5 respectively, in a tuple of the format $\langle r_j, s_j \rangle$, where r_j is the number of good interactions with a_j, and s_j is the number of bad interactions. We recall n_i denotes a_i's total number of interactions.

In our ride sharing example, an interaction between two agents represents a driver giving a passenger a lift. We assume that all drivers are continually giving lifts and being paid at the same rate. Utility represents how pleasant the interaction experience was, which may include but is not limited to, the passenger not producing full payment, or a passenger giving a tip.

At the end of a time step, when all interactions have occurred, trustees leave the network with a probability P_{te}, and trustors with probability P_{tr}. They are replaced by a new agent of the same profile, thus maintaining

the distribution of agent behaviours in the population. Burnett *et al.* used a high value for P_{te} to highlight how established agents could use their experiences with other trustees to identify the behaviour of new agents. In this paper, our evaluation considers a higher P_{tr}, to emphasise how new agents can choose a partner with no experience of the correlations between tags and behaviour.

4 Bootstrapping Trust

Probabilistic trust assessment uses personal experiences to calculate the expected behaviour of another agent. When combined with subjective logic, this assessment includes an *a priori* which is weighted higher when the probabilistic trust assessment is uncertain. BRS uses a default value of 0.5 as the *a priori* for all agents. This section describes how the *a priori* can be replaced using other information about agents e.g observable features. We first describe the probabilistic trust algorithm, and how a stereotype model can bootstrap trust, which we refer to as the SB model. Finally, we describe how tags can bootstrap both trust and stereotypes, which we refer to as the TSB model.

4.1 Trust and Reputation

Recall that agent a_i maintains a tuple, $\langle r_j, s_j \rangle$, recording the number of good and bad interactions with agent a_j. To estimate agent a_j's behaviour from this history we use beta probabilities. First, we calculate the *belief*, b, which represents the expected behaviour based solely on the direct experiences. Using subjective logic, we weight the *a priori*, a, by an *uncertainty* factor, u, which decreases as more direct experiences are collected. The overall expected behaviour, *trust*, is calculated according to Equation 8 [JI02, BNS10].

$$b = \frac{r_j + 1}{r_j + s_j + 2} \tag{6}$$

$$u = \frac{2}{r_j + s_j + 2} \tag{7}$$

$$trust = b + u \times a \tag{8}$$

If we do not assume honest reputation providers, then agents might lie about their experiences, similarly when agents use biased behaviours one trustee will treat two trustors differently, and so a recommendation may not be applicable. It is therefore important to weight the recommendations from third parties.

Equation 9 aggregates evidence about a trustee a_j from third party trustors, by replacing the values of r_j and s_j in Equations 6, 7 and 8 with the sum of good and bad experiences with the target trustee from all the trustors [JI02]. Honesty is assumed in the SB model [BNS10], and so all the opinions from trustors are weighted equally. However in the TSB model it is possible to evaluate another trustor by comparing tags, despite never having interacted with them (because trustors do not interact with other trustors). This is particularly advantageous when agents use biased behaviours because a similar trustor would likely be treated the same by a trustee, and so we should weight that trustor's opinion higher.

$$r_j = \sum_{tr}^{\mathcal{A}_{tr}} r_j^{tr} \times w, \quad s_j = \sum_{tr}^{\mathcal{A}_{tr}} s_j^{tr} \times w \tag{9}$$

Where the weighting, w, of a trustor's opinion is 1 if the SB model is being used. Otherwise, if the agent is using the TSB model, w is calculated by how similar the two trustors are, as in Equation 1. A trustor will weight the reputation information provided by an observably similar trustor higher, as the trustee is likely to treat them the same.

4.2 Stereotypes

The results of interactions used in trust algorithms are associated with specific agents. Conversely, stereotypes assume that data from one agent can have similar implications to a different agent with similar observable features. Therefore, stereotypes associate the results of interactions with observable features instead of with individual agents.

The learning interval L, is the minimum number of experiences an agent must have to build the stereotype model, and it is rebuilt every L interactions with the data accumulated in that time. The data has the form

$\langle \overrightarrow{\tau}, trust \rangle$ which is used to train an M5 decision tree. The tree learns correlations between tag values and the class variable, trust, which is calculated using Equation 8.

The decision tree learns correlations between features and behaviour so that when a new agent is encountered, their tags are classified by the decision tree and an expected behaviour is output. The output of the decision tree replaces the *a priori* value of Equation 8. Before this time, the default value of 0.5 is used. As interactions with a specific agent are collected, the *uncertainty* will decrease, allowing the trust algorithm to take more precedence over the stereotype model for that agent.

The TSB model extends this by replacing the *a priori* before L experiences are collected, as explained in Section 4.3. However, once the agent can build a stereotype model this takes precedence over the tags.

4.3 Bootstrapping Using Tags

The stereotype model cannot be built until $n_i \geq L$, and so before this we replace a in Equation 8 with the outcome of the following tag-based model. This solution does not incur any serious space or time constraints as it scales with the number of tags. A trustor maintains Q-values for each tag at different levels of similarity. These estimate the trust for agents whose tag has that level of similarity to their own. Using Q-learning prevents ignoring experience information up until L interactions have occurred. Error calculations for each Q-value are used in an attempt to identify irrelevant features early on. We measure similarity using distances between values of the various tags.

Each tag is one of the types described in Table 1. Different types of tags require different distance calculations, examples of which are given in this table. The output of each distance calculation is in the range $[0, 1]$.

$$x \leftarrow \left\lfloor dist(\overrightarrow{\tau_t^i}, \overrightarrow{\tau_t^j}) \times k \right\rfloor \tag{10}$$

Agents maintain two $k \times |\overrightarrow{\tau}|$ matrices: one containing Q-values for trust estimates and one for the average error for each Q-value over time. The parameter k specifies how many equal-width bins to discretize the output of the continuous distance function between tags into. The matrices are then indexed by each tag at each distance interval. When trustor a_i is considering trustee a_j's t^{th} tag of, $\overrightarrow{\tau_t^j}$, it will fall into bin x given by Equation 10, which scales the output of the tag's distance function. The value of k will have a relationship with the number of profiles, but agents do not know how many profiles exist and so each agent is assigned k at random between 0 and 10.

The first matrix, M^q, maintains Q-values for each tag t in distance bin x at $M^q[x][t]$. The Q-values are learnt from the updated trust values after an interaction has occurred as this is the only information available to them. Trust is an expectation of behaviour and so the Q-values are learning expected behaviour indirectly for each <tag, distance> state. The update equation for this reinforcement learning can be seen in Equation 11, taken from [Wak92]. We set the discount factor, γ, to 1 so as to weight all past experiences equally, and the learning rate, λ that determines the effect of an update on the new Q-value is 0.3. Different values for λ were not found to significantly vary the results, therefore this is not explored further.

The second matrix, M^ϵ, maintains at index $M^\epsilon[x][t]$ a running average error between the original Q-value at that index, and the newly calculated trust after the interaction has occurred. This calculation is in Equation 12, where m_i^t is the number of times this tag-distance pair has been updated in order to calculate an average. This is attempting to identify irrelevant features as a continually high error for certain tags implies it is likely unrelated to behaviour.

$$M^Q[x][t] \leftarrow M^Q[x][t] + \lambda(new\_trust - \gamma(M^Q[x][t])) \tag{11}$$

$$M^\epsilon[x][t] \leftarrow M^\epsilon[x][t] + \frac{|trust - M^Q[x][t]| - M^\epsilon[x][t]}{m_i^t} \tag{12}$$

Using these two matrices, we calculate the *a priori* trust using a weighted sum as given in Equation 13. The intuition is that each tag is associated with an expected reward, and we weight the influence this has on the overall expected behaviour calculation by the error associated with it.

$$a \leftarrow \frac{\sum_t^{|\overrightarrow{\tau}|} M^q[x][t] \times (1 - M^\epsilon[x][t])}{|\overrightarrow{\tau}|} \tag{13}$$

Table 4: Default environment parameters

parameter	value
# rounds	200
# trustors	20
# trustees	500
# relevant tags	6
# irrelevant tags, N_{nf}	6
trustee turnover, P_{te}	10%

The values in matrix M^Q belonging to trustor a_i are initialised to how a_i would behave towards theoretical partners with each of those tags at each of the distance intervals from their own tags. With the simple behaviours, this means all tags in every interval are initialised to μ_i. When behaviours are biased the same tag will be initialised to a less cooperative value the larger the distance between them are. The error matrix is initialised to 0, giving all tags an equal weighting of 1 to begin with as we do not know which are irrelevant.

5 Evaluation and Results

The results in this section compare the tag-bootstrap model (TSB) described above, to the original stereotype-bootstrap model (SB) from [BNS10]. The parameters described in Section 3 remain static throughout the experiments and have the values in Table 4. The parameters that vary in our experiments are: the probability of trustors leaving the network, P_{tr}, the proportion of an agent's lifetime they should explore for, α, and the learning interval, L.

The aim of this work is to aid agents in establishing themselves as quickly as possible by improving their rewards in the early stages of their life. The metric of success for the trustors is their average outcome over time, which will indicate whether they have chosen good partners. An important difference between biased behaviours and simple behaviours are that the mean values defining agent behaviours are different between the same profiles, therefore the results are not directly comparable between simple and biased behaviours and only the trends can be compared. In the following results, biased behaviours appear to cause less average utility over time, but this is due to the average outcome of a p_1 trustee being less when they are interacting with dissimilar trustors. The best agent profile type is still being chosen for interactions.

5.1 Simple and Biased Behaviours

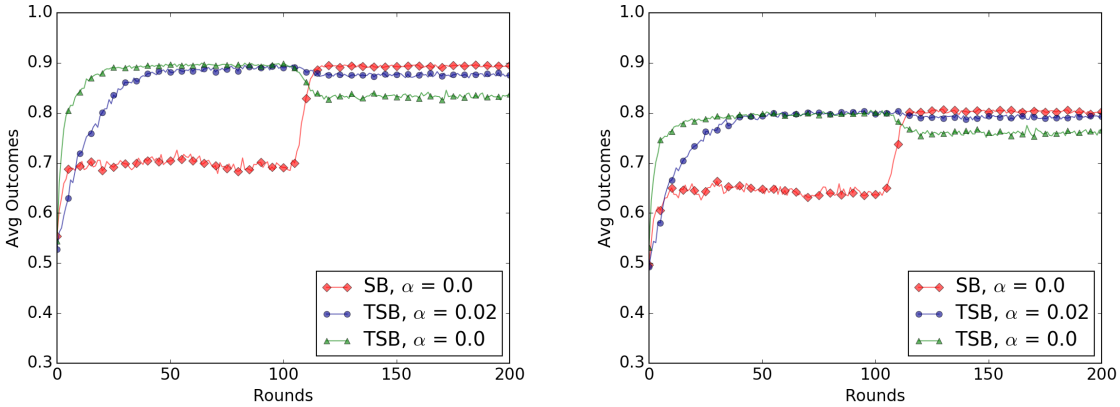
We compare the results of the SB model from Burnett's work to our TSB model as it is an extension of the former [BNS10]. The first results in Figure 3a demonstrate a comparison to the original work in the closest like-for-like environment[1]. The initial gain in outcome is clearly visible, however when there is no exploration in the TSB model the average outcome reduces once the stereotype model is built (around the 100th round)[2]. We identify the source of this in the next section on exploration. The tag subsystem is also vulnerable to noisy features in early stages. When this data trains the decision tree it produces less accurate predictions and thus why it performs worse later in time. By forcing agents to explore a small amount initially, this already improves results, as can be seen in both Figures 3a and 3b. The best value for exploration, α, is 0.02 as demonstrated in Section 5.2.

Recall that the absolute value for average outcome from simple behaviours are not comparable to biased behaviours, because the means defining trustee behaviour vary with similarity. When trustees use simple behaviours and all trustors are interacting with p_1 trustees, the average outcome will be the μ associated with p_1. However, when μ varies and all trustees being chosen for interaction of type p_1, the average outcome will depend on how p_1 trustees act with all the different types of trustors. The results converge on 0.8 when all trustees chosen are p_1. Therefore analysis of the results compares the trends.

When agents have biased behaviours, the trends are very similar to simple behaviours, as seen in Figure 3b. This demonstrates that despite inclinations to interact with similar agents, the TSB model can still identify the best agents to interact with.

[1] Specifically, we compare the results of global interaction and reputation sources (GGS) from the work of Burnett *et al.*

[2] All results in this paper were proven to be statistically significant using t tests with the area under the curve to remove temporal dependencies over 50 runs for $p < 0.001$.



(a) Agents use simple behaviours.

(b) Agents use biased behaviours

Figure 3: Comparison of the SB and TSB models

5.2 Model Accuracy

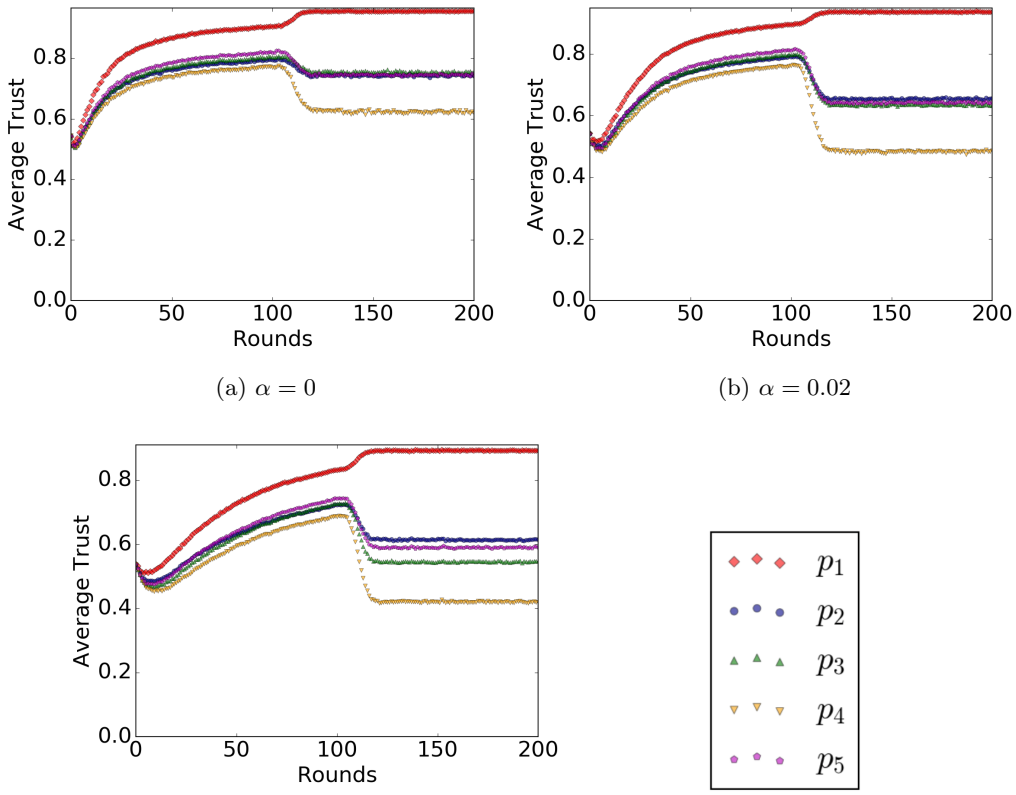
With no exploration, the TSB model causes the decision tree trained afterwards to perform worse than when the TSB model is not used. This is because the TSB model quickly identifies the more extreme behaviours i.e agents of profiles p_1 and p_4, because the extreme profiles consistently perform either good or bad given our binary outcome scale for an interaction. Without exploration, the TSB model always selects p_1 agents for interactions, never having enough interactions with agents of other profiles to accurately learn the expected behaviour or the tags which correlate with them. The decision tree is then trained on this inaccurate data, and as a result cannot easily distinguish between the remaining profiles. Figure 4 confirms this, showing trustors' average estimated behaviour of the five profiles using the TSB model. With no exploration, as seen in Figure 4a, the four worst profiles are estimated to have almost identical behaviour by the stereotype model, and only one direct experience will influence which partner to choose regardless of how representative that one interaction is of true behaviour. As exploration increases, in Figures 4b and 4c, the TSB model has more interactions with other profile types and thus gathers more data on them enabling the decision tree to distinguish between profiles more clearly. However, this long term improvement incurs an earlier cost, as the agents are not able to interact with profile one agents when they would otherwise choose to.

Exploration is therefore an important consideration of the model, and different environments will require more or less exploration. If the trustors are not able to leave the population then they have more time to explore. Figure 5 shows that the SB model performs the same regardless of exploration, this is because its partner choices are initially random, and so the exploration has little effect. Having identified that more exploration leads to a more accurate decision tree later in time, we now confirm the effects on short term utility of this. A high α of 0.1 results in the best decision tree, but slowest gain in utility early on. The best compromise is when α to be 0.02, because there is minimal long term utility loss as a result of a less well trained decision tree, but the short term utility compared to no exploration is still high.

When the trustor turnover is marginally increased to 0.01, it is more beneficial for agents to identify the best partners despite not accurately assessing their behaviour, as they leave the system before needing any estimation of agent behaviour. Figure 6 shows the effects of exploration in such a dynamic population, and the TSB model with low exploration continuously does better. The appropriate exploration is very dependent on the environment which agents are exploring. The advantage of using α is it varies decay as a proportion of agent lifetime, so we can intuitively see how to include it in other evaluative scenarios.

5.3 Dynamic Population and Learning Interval

The SB model performs well with both low or high values of L because the relationship between tags and behaviour is static so old interaction histories are always representative of current agents' profiles. In our ride sharing example, relevant tags which correlate with behaviour that might change include location (e.g. a new nightclub in a previously quiet area). In this case, methods relying on historical data such as stereotyping, will be slower to learn the new correlation compared to the tag based method. We are not implying that tags can be



(a) $\alpha = 0$

(b) $\alpha = 0.02$

(c) $\alpha = 0.1$

(d) Key

Figure 4: Average trust in profiles, agents use simple behaviours, TSB

falsified, but that correlations between tag values and behaviours might change. The TSB model will not be as susceptible to this change, because an agent's own tags and behaviours will have changed too, for example a taxi driver sharing the same location tag as the new nightclub will be aware of the change. Despite not demonstrating this dynamic behaviour, we present results for both simple and biased behaviours in Figure 7. The TSB model performs at least as well as the SB model for any value of L.

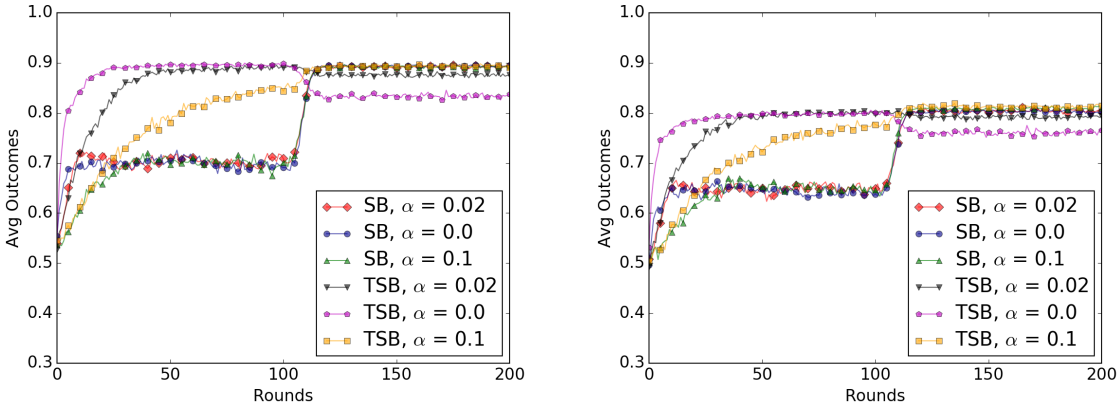
When P_{tr} is 0.1 an agent's average lifetime is 10, Figure 7 verifies that when L is 100, the SB model performs poorly because agents are solely relying on personal experiences to calculate expected behaviour. However, when L is 1 they have the chance to learn correlations between features and behaviour and start to perform better. The TSB model results presented here performs at least as well as the best SB model because it is accurately choosing agents having profile p_1, so maximising outcome.

Environments ranging in complexity will have different optimal values of L to maximise learning without including outdated information. However, the TSB model reduces the need to know this in advance, which is especially important where the trustor turn over is high as trustors might not ever get the chance to collect L experiences to build their stereotype model.

5.4 Inaccurate Reputation Sources

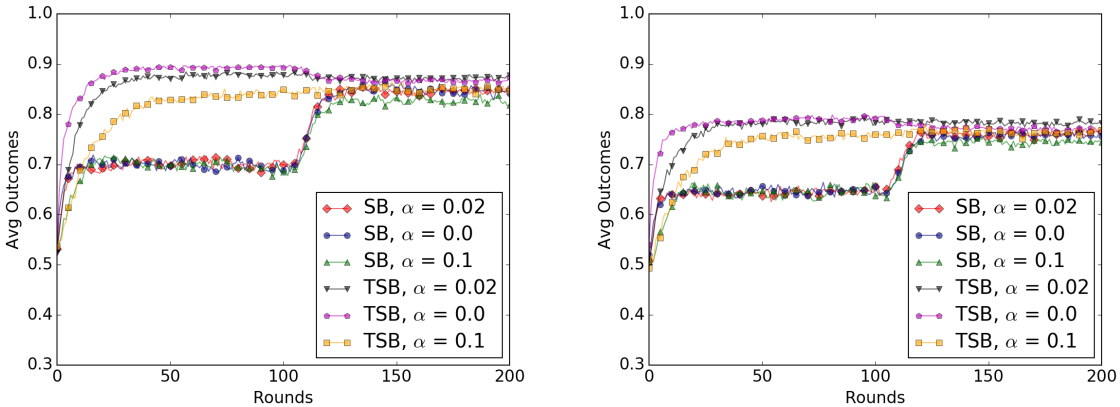
Trustors might give inaccurate accounts of interactions with trustees for two reasons. Firstly, when agent behaviours are biased towards similar agents, a trustee will treat one trustor differently to another if those two trustors are dissimilar. Secondly, the trustor could be lying for their own selfish motivations. We expect the TSB model to account for the former issue by weighting feedback from similar agents higher, as described in Equation 9. With the latter case, both models are susceptible to this noise unless they use a more advanced reputation model such as TRAVOS [TPJL06]. Therefore, we evaluate in an environment where a third of sources report exactly the opposite of their experience, a third are noisy sources, and a third are accurate.

The results in Figure 8 show that both models perform similarly to when agents are honest. With such a high trustee turnover, P_{te}, direct experiences are rarely weighted much higher than the *a priori* of the model



(a) Simple Behaviour (b) Biased Behaviour

Figure 5: Different exploration rates, α, for $P_{tr} = 0$.



(a) Simple Behaviour, (b) Biased Behaviour

Figure 6: Different exploration rates, α, for $P_{tr} = 0.01$.

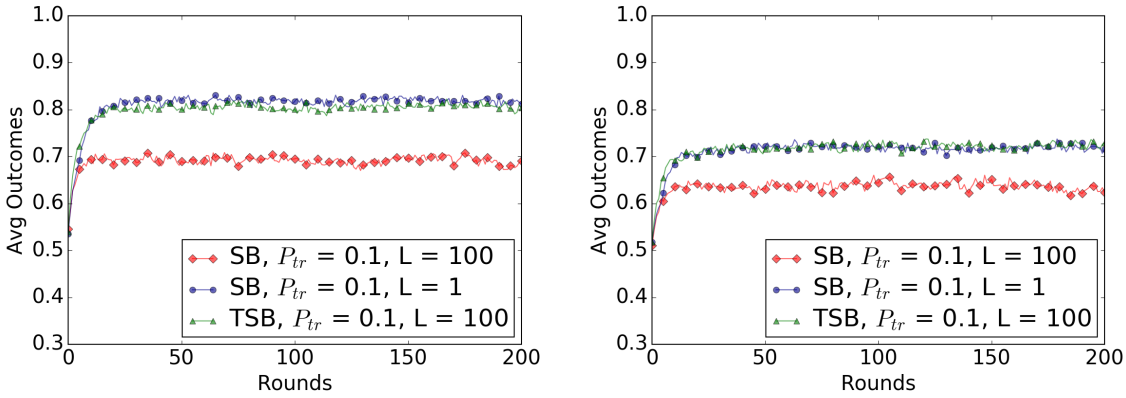
because they are not available. As the SB model uses the default *a priori* for all agents, experiences are the only distinguishing feature between trustees, making it more susceptible to lying reputation sources. We can see this in the early stages of the SB model. However, the TSB model uses the trustees features to calculate an *a prior* in the early stages, and so converges at the best average outcome when sources are lying, just slower than when they honest.

6 Discussion and Conclusions

Tags are observable features inspired from evolutionary studies showing similarity encourages selective altruism. We have shown that when agents use the concept of *relatedness*, they have the ability to identify cooperative interactions partners without external information such as prior experiences with the agent. Using this model to bootstrap existing trust and stereotype models improves an agent's reward in the early stages of its life. This is especially prominent in dynamic populations where agents have insufficient experiences to successfully use other trust and stereotype techniques.

The model presented in this paper accurately identifies extreme behaviours, which is useful to agents who do not have enough time in the network to accurately predict behaviours of all profile types. However, this can produce inaccurate long term estimations of behaviour for all profile types which we have shown can be alleviated with exploration. A trade off must be made between how much agents should explore against immediate short term utility gains.

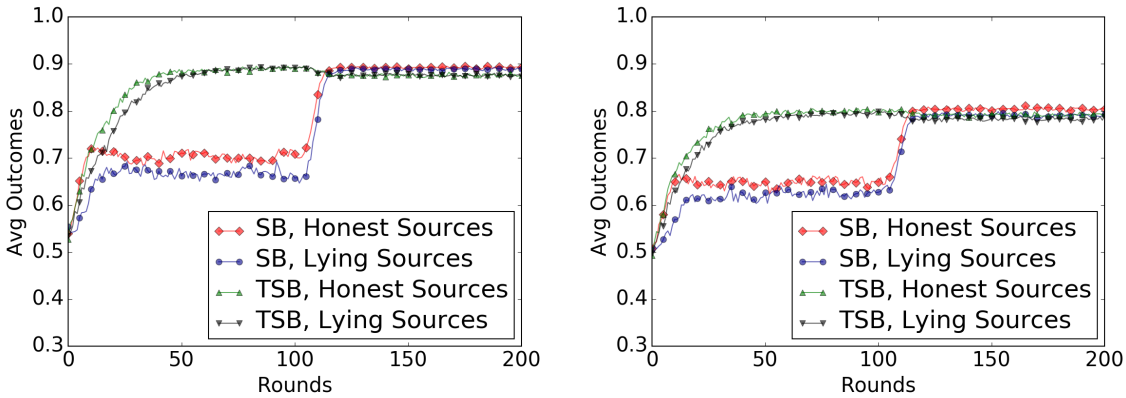
The amount of exploration depends on the context. If agents can afford to establish themselves slowly and



(a) Agents use simple behaviours.

(b) Agents use biased behaviours.

Figure 7: Comparison of learning intervals, L, with $P_{tr} = 0.1$



(a) Agents use simple behaviours.

(b) Agents use biased behaviours.

Figure 8: Coping with inaccurate reputation sources.

so with exploration, the short term utility gain is reduced but improves the accuracy of the stereotype model being bootstrapped. However, in competitive and dynamic environments it is often important to perform well as quickly as possible. This is called "survival of the fittest" in evolutionary theory and is prevalent in modern society, for example competition in market places [Hun97]. The tag system presented here offers quick and large rewards to new agents with less exploration.

Future work includes allowing the tag-based model to take precedence even after enough experiences have been gathered for a stereotype model. If agent behaviours are dynamic, changing which tags are indicative of good behaviour over time then the stereotype model will have trained on unrepresentative data. Reverting to tag comparisons could prevent ages from choosing bad partners using old data.

One final consideration for future work is handling cheating agents. In related literature these agents misrepresent themselves by mimicking tag values which are indicative of good behaviour. Cheaters will be selected for interactions but then behave badly and selfishly. This is common in real world applications such as online marketplaces or peer-to-peer networks, for example as fraud or virus spreading. A limitation to the work presented here is that tags can be easily mislead, though the inclusion of BRS makes it less susceptible than solely tag-based models.

7 Acknowledgements

This research is funded by the EPSRC.

References

[AD+88] Robert Axelrod, Douglas Dion, et al. The further evolution of cooperation. *Science*, 242(9):1385–1390, 1988.

[AHG04] Robert Axelrod, Ross A. Hammond, and Alan Grafen. Altruism via kin-selection strategies that rely on arbitrary tags with which they coevolve. *The Society for the Study of Evolution*, 58(8):1833–1838, 2004.

[Axe84] Robert Axelrod. The evolution of cooperation. *Basic Books, New York. Econornetrica*, 39:383–396, 1984.

[BNS10] 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.

[BNS13] Chris Burnett, Timothy J Norman, and Katia Sycara. Stereotypical trust and bias in dynamic multiagent systems. *ACM Transactions on Intelligent Systems and Technology*, 4(2):26, 2013.

[Daw76] Richard Dawkins. *The Selfish Gene*. Oxford University Press, 1976.

[Gam00] Diego Gambetta. Can we trust trust? *Trust: Making and breaking coooperative relations*, 13:213–237, 2000.

[Ham64] W. D. Hamilton. The genetical evolution of social behaviour. *Journal of Theoretical Biology*, 7(1):1–16, 1964.

[HE03] David Hales and Bruce Edmonds. Evolving social rationality for MAS using tags. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 497–503. ACM, 2003.

[HJ04] Trung Dong Huynh and Nicholas Jennings. Fire: An integrated trust and reputation model for open multi-agent systems. In *ECAI 2004: 16th European Conference on Artificial Intelligence, August 22-27, 2004, Valencia, Spain: including Prestigious Applicants [sic] of Intelligent Systems (PAIS 2004): proceedings*, volume 110, page 18, 2004.

[HO05] Enda Howley and Colm O'Riordan. The emergence of cooperation among agents using simple fixed bias tagging. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 2, pages 1011–1016. IEEE, 2005.

[Hun97] Shelby D Hunt. Resource-advantage theory: An evolutionary theory of competitive firm behavior? *Journal of Economic Issues*, 31(1):59–78, 1997.

[JI02] Audun Jøsang and Roslan Ismail. The beta reputation system. *Proceedings of the 15th Bled Electronic Commerce Conference*, 5:2502–2511, 2002.

[LDRL09] 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*, pages 7–16, 2009.

[RCA01] Rick L Riolo, Michael Cohen, and Robert Axelrod. Evolution of cooperation without reciprocity. *Nature*, 414(6862):441–443, 2001.

[SS01] Jordi Sabater and Carles Sierra. Regret: reputation in gregarious societies. In *Proceedings of the fifth international conference on Autonomous agents*, pages 194–195. ACM, 2001.

[ŞYN16] Murat Şensoy, Burcu Yilmaz, and Timothy J Norman. Stage: Stereotypical trust assessment through graph extraction. *Computational Intelligence*, 32(1):72–101, 2016.

[TLRJ12] Luke Teacy, Michael Luck, Alex Rogers, and Nicholas Jennings. An efficient and versatile approach to trust and reputation using hierarchical bayesian modelling. *Artificial Intelligence*, 193:149–185, 2012.

[TPJL06] Luke Teacy, Jigar Patel, Nicholas Jennings, and Michael Luck. Travos: Trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multi-Agent Systems*, 12(2):183–198, 2006.

[Wag10] Alan R Wagner. Using stereotypes to understand one's partner. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, volume 1, pages 1445–1446. International Foundation for Autonomous Agents and Multiagent Systems, 2010.

[Wak92] Christopher John Cornish Hellaby Wakins. Q-learning. *Machine Learning*, 8(3-4):279–292, 1992.

[YCS15] Osman Yucel, Chad Crawford, and Sandip Sen. Evolving effective behaviours to interact with tag-based populaitons. *Connection Science*, pages 1–17, 2015.