Addressing Class Imbalance in Trust and Stereotype Assessment

Caroline Player and Nathan Griffiths

Department of Computer Science
University of Warwick
United Kingdom
c.player@warwick.ac.uk
n.griffiths@warwick.ac.uk

Abstract. Trust, reputation and stereotypes enable agents to identify reliable interaction partners based on past interactions. However, such methods can cause agents to choose the same known partners instead of unknown, but potentially better, alternatives, giving rise to a class imbalance in their interaction histories. In this paper, we present a Class Imbalance Modification (CIM) method, to improve agents' initial assessments of others by becoming aware of the bias towards known agents. CIM enables an agent to determine whether data-driven trust, reputation and stereotypes are appropriate to assess a target agent, depending on how representative the agent's past interaction data is of the target. We also present a technique, Direct Comparative Stereotypes (DCS), which does not use past interaction data to make a stereotypical assessment, and so can be used if CIM concludes the data is inappropriate. Finally, CIM determines whether data-driven models have been rendered inappropriate by dynamic agent behaviour, where old interactions may no longer reflect current behaviour. Our results show that CIM significantly reduces error in *a priori* estimates, which improves partner selection and increases average utility.

Keywords: Trust; Reputation; Stereotypes; Class Imbalance; Multi-Agent Systems

1 Introduction

Agents in a MAS typically have different abilities, priorities and motivations, and so identifying an agent to achieve a particular task is challenging. In decentralised contexts such as MAS, security protocols can be unenforceable because they are not scalable, the connected technologies are too different or there is a lack of central authority [6, 8, 18]. Social mechanisms such as trust, reputation and stereotypes, are alternative approaches to guide agents when deciding who to interact with. Trust and reputation systems use past experiences with a target agent to assess the probability they will achieve the task [10, 14], while stereotype algorithms enable such assessments where no such past experiences exist. Stereotypes exploit the assumption that a set of observable features exist which correlate with agents' behaviour [5]. Stereotype methods can bootstrap trust and reputation algorithms by providing initial assessments when past experiences are not available.

Most trust and reputation algorithms are able to identify when the necessary data (experience data with a target agent) is not dependable. In this case, a default assessment, or a stereotype assessment if it is available, can be used. A similar problem exists in stereotype models if the past interaction data agents used to build the model does not represent the type of agent currently being evaluated, however, existing stereotype models do not account for this. This *class imbalance* in the history of past interactions is caused by the realistic assumption that agents are self interested, and make decisions based on their local knowledge to select who they believe are the most trustworthy partners. However, once an agent has interacted with another that they know to be more trustworthy than the default trust value ascribed to unknown agents, they continue to interact with them and never expand their knowledge. This prevents agents from accurately learning a variety of stereotypes, and ultimately from discovering more trustworthy partners.

Dynamic behaviour is also challenging for stereotype algorithms. Due to the autonomous, decentralised nature of MAS, agent behaviours can change at times and rates unknown to others. This is problematic for stereotype models which predict future behaviour based on past data. Our work is evaluated with both static and dynamic behaviours.

In this paper, we present a Class Imbalance Modification (CIM) method to compliment stereotype models, which considers if an assessor's history does not represent the agent being assessed. Second, we present a data-free stereotype technique, Direct Comparative Stereotypes (DCS), which can be used if CIM deems a data-driven stereotype technique inappropriate. DCS is inspired by tag-based cooperation where agents believe that agents who look like themselves will behave similarly [15], and aims to provide an initial estimate of behaviour. Our results show that using CIM and DCS together with trust, reputation and stereotype algorithms improves the accuracy of agents' assessments and therefore their ability to select good partners for interactions, reflected in the improved average utility they receive. Our technique also achieves a more accurate assessment of all of the agents in the population without exploration, because CIM avoids misclassifying agents that are under represented in the past interaction data.

The remainder of this paper is structured as follows. Section 2 discusses the state-of-the-art in the related fields. Sections 3 and 4 introduce CIM and DCS respectively, and the evaluation environment is described in Section 5. The results are presented in Section 6, and finally Section 7 concludes the paper.

2 Related Work

We adopt a probabilistic definition of trust, using direct experiences with others to estimate future performance [7, 9, 13, 19]. Similarly, reputation aggregates reports of experiences from multiple agents to assess a target [14, 1, 11]. The Beta

Reputation System (BRS) is a mathematically rigorous approach to trust and reputation which is adopted in much of the literature because of its generality [10]. Since trust is not the focus of this paper, we use BRS as an exemplar (which could be substituted if required). Our only requirement of a trust algorithm is that an *a priori* value contributes to the trust assessment. BRS considers past interactions between agents to have been either good or bad. A trustor agent, tr, calculates trust in a trustee, te, by combining the expected value of a beta probability function (which uses the number of good experiences, r_{te}, and bad experiences, s_{te} with te as parameters) with subjective logic to account for uncertainty. The belief in the trustee, b_{te} represents an estimate of the trustee's behaviour purely based on past experiences. The level of uncertainty, u_{te}, depends on how much data is used to calculate, b_{te}, and weights an *a priori* value accordingly. The *a priori* value, a_{te}, of trust needs to be provided, possibly by a stereotype algorithm discussed below, otherwise a default value is used. Finally, trust is calculated as follows:

$$b_{te} = \frac{r_{te}}{r_{te} + s_{te} + 2} \tag{1}$$

$$u_{te} = \frac{2}{r_{te} + s_{te} + 2} \tag{2}$$

$$a_{te} = \begin{cases} x, & \text{if a model is available} \\ 0.5 & \text{otherwise} \end{cases} \tag{3}$$

$$trust(te) = b_{te} + u_{te} \times a_{te} \tag{4}$$

A trustee's reputation is an aggregation of experiences with them from opinion providers op, such that r_{te} and s_{te} are calculated as:

$$r_{te} = \sum_{op}^{\mathcal{A}_{tr}} r_{te}^{op}, \quad s_{te} = \sum_{op}^{\mathcal{A}_{tr}} s_{te}^{op} \tag{5}$$

where $\mathcal{A}_{tr}$ is the set of trustors.

A variety of complications in real-world scenarios reduce the efficacy of trust algorithms which only use past experiences with an agent to estimate their current behaviour. One problem is that trust algorithms are inaccurate when there is no available past experiences with an agent. This problem arises when an agent first joins a system, or if the agents are connected in a dynamic network such that agents encounter new connections. Stereotype models have been designed to address this issue by assuming an agent that has never been seen before, might behave similarly to agents who are observably similar and whose trust value is already known [5]. This relies on the assumption that agents have a set of observable features, $\overrightarrow{X}$ which correlate with behaviour, Y, such that agents can learn the stereotype function $f(\overrightarrow{X}) \rightarrow Y$. Stereotype models often use machine learning methods to learn this mapping. For example, agents can train a decision tree on their history of past interactions with other agents [1]. STAGE maps the relationships between features and feature values as an ontological graph which is then mined for patterns of trustworthy and untrustworthy agents [17]. This specifically aims to overcome some limiting factors of existing models which can only learn simple correlations, and also focuses on tackling whitewashing attacks. However, this method relies on a very detailed agent structure, and a full and accurate view of that ontology. StereoTrust derives an assessment of an agent who can belong to multiple groups [11]. The focus of StereoTrust is how to aggregate multiple assessments from overlapping stereotypes instead of how the stereotypes are identified (such as selecting the observable features of agents with the highest information gain). StereoTrust has been evaluated in terms of accuracy and computation speed, but not how successful the agent ultimately is in terms of utility. StereoTrust is the only existing algorithm to consider that there may be too few instances to assess a particular group. However, since StereoTrust assumes groups are identifiable, by aggregating the data from that group it can statistically identify if this gives enough data for an accurate assessment. Alternatively, our CIM method considers whether there is sufficient data to group the agents in the first instance.

Burnett *et al.* discuss two types of stereotypical bias called perceptual bias and behavioural bias [2]. These refer to an agent either subjectively perceiving interaction outcomes differently, or agents behaving differently towards particular partners, respectively. Similarly to behavioural bias, Liu *et al.* describe how agents might only give ratings if something is good or bad, making it hard to predict when the agent will rate differently [11]. These are important issues to address, especially if agents have the ability to share stereotypical reputation. However, these are different types of bias to class imbalance, which we address in this paper.

We address the class imbalance problem where agents have information on only a few agents or agent types, caused by the realistic assumption that agents are self interested, and cannot be relied upon to voluntarily make a sub-optimal decision to gain knowledge at the cost of utility. Exploration can overcome class imbalance by forcing agents to interact with others with whom they otherwise would not [12]. However, realistically it may not be possible to force self interested agents to make this sub-optimal decision, and it may cost them in terms of utility. Class imbalance in data mining problems is widely addressed [4, 3]. However, such approaches typically address class imbalance in data analysis when there is central control and no self interested agents affecting data collection.

3 Class Imbalance Modification for Stereotypes (CIM)

CIM draws upon elements of C-DenStream, a semi-supervised, density based, online clustering algorithm [16]. A density based clustering algorithm uncovers groupings in the data without knowing how many groups may exist in advance, or given an agent may only ever encounter agents from a subset of the profiles which exist in the population. Stereotype models use fully labelled data sets regardless of the certainty in those trust labels. However, CIM uses a semi-supervised clusterer which only labels instances where the agent is highly confident in that trust value, meaning it was based on multiple repeat interactions, making it more likely to be accurate. Finally, the online component makes CIM time efficient. This is important in a MAS application as clustering results may be required frequently, or because limited data storage is available. Most stereotype models use machine learning algorithms which can only rebuild every L time steps. This is one cause of their vulnerability when agent behaviours are dynamic, as the model can be built on data representing old behaviour. We show how CIM integrates with stereotype models being built every L time steps.

An agent records an interaction as a tuple: $\langle te, \overrightarrow{\tau}, o, t \rangle$, where $\overrightarrow{\tau}$ is the observed features of te, o is the interaction outcome and t is the time of the interaction. Trust in te after the interaction is assessed with the trust algorithm, in our case BRS as defined in Equation 4. Only the observable features and trust are used to train the stereotype model, so the data is reduced to: $\langle \overrightarrow{\tau}, trust \rangle$. The stereotype model retains the trust label for all instances but CIM is semi-supervised, and only keeps the trust label if the uncertainty, u_{te}, calculated in Equation 2, is below the threshold υ, whose value does not negatively impact results, as we demonstrate in Section 6.

Algorithm 1 Agent process using CIM

```
 1: for $t \in T$ do
 2:     $partner \leftarrow te \in \mathcal{A}_{te} \mid max_{te \in \mathcal{A}_{te}} trust(a)$        ▷ Identify best partner
 3:     $data \leftarrow data + \langle te, \overrightarrow{\tau}_{te}, trust, t \rangle$
 4:     C-DenStream $\leftarrow update(\overrightarrow{\tau}_{te}, trust)$
 5:     $C_t \leftarrow$ CDBSCAN($MC_p$)        ▷ This clustering is always up to date
 6:     if $t$ mod $L$ then
 7:         build_Stereotype(data)
 8:         initialise_C-DenStream(data)
 9:         $C_s \leftarrow$ CDBSCAN($data$)        ▷ Clustering at the time of building stereotypes
10:         $C_t \leftarrow$ CDBSCAN($data$)
```

Algorithm 1 shows that for each time step t, a trustor using CIM first identifies the agent they believe has the highest likelihood of a good interaction, interacts with them and updates CIM. Line 4 encompasses that data is maintained online with C-DenStream (discussed further below). CIM maintains two local offline sets of clusters, C_s and C_t, produced by the final clustering step of C-DenStream, CDBSCAN, at different times. C_s is constructed with the same past experiences that build the stereotype model s, to address the class imbalance problem. If a trustee's features are not associated with a cluster in C_s then they are not represented by the stereotype model. C_t is reset at the same time as C_s but also updated every time step with new interaction data so that it is up to date, t. Comparing C_t with C_s addresses dynamic behaviour because the clusterings will be different if behaviour has changed. C-DenStream uses a forgetting factor, λ, to give precedence to new data.

C-DenStream creates micro-clusters with the raw data in the initialisation step, followed by an offline CDBSCAN step with the micro-clusters output from that step to produce C_s and C_t on lines 9 and 10 in Algorithm 1. The *micro-clusters* are small spherical clusters that summarise raw data. Micro clusters have a weight which is increased as new instances are added to it, and fades over time. If the weight of a micro-cluster is above a threshold it is a potential-micro-cluster *p-mc*, else it is an outlier-micro-cluster, *o-mc*. An *o-mc* might grow into a *p-mc* over time, and a *p-mc* can fade into an *o-mc*. Forming micro-clusters requires the parameters ϵ, β and μ representing the neighbourhood radius, the outlier radius and the minimum number of points in a neighbourhood, respectively. We set these to 1.8, 1, 3 to create small micro-clusters in our evaluation where there are 7 observable features, each with a maximum value of 5. These parameters create a *p-mc* of size at least 3, otherwise it is an *o-mc*. Final clusterings for C_s and C_t are a set of grouped p-micro-clusters created with CDBSCAN. Line 4 updates the micro-clusters online according to C-DenStream, so that C_t can be updated every time step efficiently by requesting a CDBSCAN clustering using the online maintained p-micro-clusters, MC_p, as input instead of raw data.

As part of building and maintaining micro-clusters with a semi-supervised algorithm, Cannot-Link constraints are enforced between labelled data which is transformed to constraints between the micro-clusters each instance belongs to. The label is the trust value binned into one of 10 bins, and if no trust value was provided because the certainty in it was not high enough, the label is -1. Points and micro-clusters need to be *label consistent* amongst members of the same cluster, meaning they are either labelled with the same trust value, or one is labelled and the other is not. Once a labelled point is added to a previously unlabelled micro-cluster, the micro-cluster is labelled and a label weight, l_w, is initialised for the micro-cluster. If an instance of the same label is added to the micro-cluster, the label weight is increased by one, otherwise it fades according to the forgetting function: $l_w = l_w * \lambda^{t-t'}$, where t is the current time, t' is when l_w was last updated, and $\lambda \in [0, 1]$ is a forgetting factor. Forgetting factors appear in the stereotype and clustering algorithms used in this work, and therefore we use the same value and notation for it. If few, or no, labelled instances arrive, then l_w will fall below a threshold, calculated in the same way as a micro-cluster weight threshold for being deleted described in the literature, and the micro-cluster will lose its label. We do not enforce Must-Link constraints because if two agents are assessed to have the same trust value this does not imply they are guaranteed to belong to the same group, as either the trust could be miscalculated or two different groups have the same behaviour at the time.

CIM is described in Algorithm 2, which is run over timesteps t for T total time. It uses C_s and C_t to decide if a trustee should be assessed with the stereotype model.

Algorithm 2 CIM

```
$label_s \leftarrow Cluster(\overrightarrow{\tau}_{te}, C_s)$
**if** no cluster found **then return** False        ▷ This accounts for class imbalance
$label_t \leftarrow Cluster(\overrightarrow{\tau}_{te}, C_t)$
**if** $label_s \neq label_t$ **then return** False        ▷ This accounts for dynamic behaviour
**return** True
**function** CLUSTER($\tau$, Clustering C)
    $label \leftarrow -1$
    **for** Cluster c $\in C$ **do**
        **if** $dist(\overrightarrow{\tau}, c_{center}) < \epsilon$ **then**
            $label \leftarrow c_{label}$
    **return** $label$
**end function**
```

4 Direct Comparative Stereotypes

DCS provides an initial assessment of a trustee without using the trustor's past interaction data, and can be used when no stereotype model exists or when CIM has assessed the stereotype model to be inappropriate for assessing the trustee. DCS compares an agent's own features to the trustee's, and if they are similar enough, the trustor assumes the trustee has the same behaviour as well. We assume an agent has awareness of its own observable features, though not which are relevant or irrelevant, and its own behaviour. DCS has two advantages, first, it does not require any past experience data so can be used immediately by new agents, or for agents whom no one knows anything about. Second, it does not rely on information from reputation, which may come from lying or biased sources.

The similarity between two agents can be calculated using a distance metric between each value in the set of observable features. Using a Euclidean distance metric, the similarity, $sim_{tr,te} \in [0, 1]$, between a trustor, te, and a trustee, te, with observable features of length, n, can be calculated as:

$$sim_{tr,te} = \sqrt{\sum_{i=0}^{n} (|\overrightarrow{\tau^{i}_{tr}} - \overrightarrow{\tau^{i}_{te}}|)^2} \tag{6}$$

If $sim_{tr,te}$ is within a threshold of similarity, δ, the trustor infers that the trustee will have the same behaviour, and this replaces the *a priori* value of Equation 3. We use $\delta = 0.85$ as a generic high value of similarity.

$$a_{te} = \begin{cases} bhv_{tr}, & \text{if } sim_{tr,te} < \delta \\ 0.5, & \text{otherwise} \end{cases} \tag{7}$$

DCS only gives a trustor insight into a small subset of trustees but if this encourages interactions with good agents, or avoids bad agents, the resulting interaction outcomes are propagated by the reputation algorithm to the benefit of all trustors.

5 Evaluation

A set of agents, is divided into disjoint subsets of trustors, $\mathcal{A}_{tr}$, and trustees, $\mathcal{A}_{te}$. Trustor agents assess available trustees to find an interaction partner. The represent a dynamic population, where each trustee has a probability, $p^{te}_{leave} \in p_{leave}$, of leaving the population each round and being replaced by a new agent.

Agents have a set of relevant observable features and a behaviour, whose values depend on which profile they belong to. Feature values can represent categories or numeric values for characteristics of agents in the applications for example, agents in an online marketplaces might have observable features such as prices, items sold, profile picture features and location amongst others. We use numerical values to abstract these feature values. Agents also have observable irrelevant features which do not correlate with behaviour and it is part of the learning task to identify these. Unlike existing work, we relax the assumption that the relevant observable features are exactly the same for all agents of a profile, as there could be a range of values or a distribution over possible values. We define a feature f as $f : \langle \mu_f, \Theta \rangle$, where $\mu_f > 0$ is the mean value of the feature and Θ is the standard deviation. As Θ increaeses the feature value becomes noisier, but we use $\Theta = 0.2$ in our evaluation to add just a small amount of noise. Profile relevant features and behaviours are generated randomly for each experiment.

For example, a profile, p, of 5 relevant features, may be defined with the feature vector: $p : \langle f_0 : 2, f_1 : 0, f_2 : 3, f_3 : 1, f_4 : 5 \rangle$. And an agent, j, belonging to this profile, where $\Theta = 0.2$ and who also has 2 irrelevant features generated randomly, might have the following observable feature vector: $\overrightarrow{\tau_j} : \langle f_0 : 2.05, f_1 : 0.1, f_2 : 2.97, f_3 : 1.17, f_4 : 4.87, f_5 : 4.1, f_6 : 0.6 \rangle$.

The behaviour associated with a profile can be static or dynamic depending on the evaluation. If behaviour is static, a profile is assigned a behaviour $bhv_p \in [0, 1]$ indicating the probability in an interaction they will behave well. If behaviour is dynamic, $bhv^t_p \in [0, 1]$, this value can change over time according to some function. To achieve this, we define a random number of static behaviours, which last a random length of time, and are transitioned to at a random speed. For patterned behaviour, a cyclical function could be used.

All the relevant variables in this paper are summarised in Table 1.

6 Results

We present the results of bootstrapping Burnett's decision tree stereotype model [1] with CIM and/or DCS. Other stereotype models could be substituted for the decision tree. All results are statistically significant for $p < 0.001$ using a paired t-test. The optimal results presented are a gods eye view of the interaction results if agents selected the true best available partners.

CIM improves the average utility agents receive, implying they selected partners with a better behaviour more often, demonstrated in Figure 1. These results show how using DCS and CIM together give the best overall results, but DCS offers more benefit when behaviours are dynamic, implying CIM identified the

Table 1. Summary of Parameters

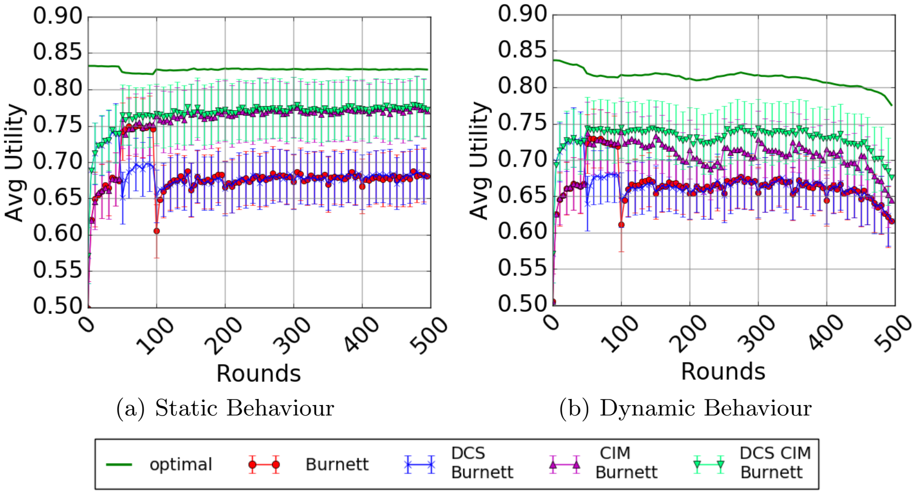
Parameter	Definition	Value
$\|\mathcal{A}_{tr}\|$, $\|\mathcal{A}_{te}\|$	number of trustors and trustees, respectively	20, 80
Θ	max standard deviation in relevant feature values	0.2
nrf, nnf	number of relevant and irrelevant features respectively	5, 2
$\mathcal{N}_{profiles}$	number of profiles	5
λ	forgetting factor. Same value applies in trust, reputation, stereotypes and clustering	0.95
p_{leave}	probability an agent is replaced by a new agent	0.1
L	number of instances to have before building stereotype model and initialising C-DenStream, respectively	50
δ	threshold of similarity for DCS	0.85
υ	minimum uncertainty threshold in trust assessment to apply a CL constraint	0.2

data had changed and it reverted to using DCS when necessary. A trustor using DCS will make the most up to date assessment of an agent's current behaviour, if they are similar enough, because if their behaviours changed then they both experienced it. For example, if two agents are in a similar location which temporarily undergoes a signal fault prompting their ability to communicate and achieve tasks to change, they would both be knowledgeable of that change without needing to learn from repeated interactions.

DCS is especially benefical in the first L timesteps (with or without CIM) before the stereotype model or CIM are initialised, because DCS is better than a default assessment. Even though DCS will only provide estimates for a small subset of trustees to one trustor, the few interactions had as a result of DCS by all trustors in the early time steps will improve good agents' reputation, allowing all trustors to become aware of good agents. After L timesteps, and when CIM is not being used, there is no dependence on DCS because there is no trustor turnover, so once trustors initialise their stereotype models they cannot choose to revert to DCS.

The accuracy of agent behaviour assessment is an important factor towards how successful an agent is at identifying partners. Our work focuses on improving the *a priori* estimate from the stereotype model, therefore, Figure 2 presents the Root Mean Squared Error between an agent's true behaviour and the *a priori* estimate of it. The stereotype model and clusterings are rebuilt every L instances, therefore periodic changes in error every 50 timesteps can be seen.

Figure 3(a) shows that as υ increases, agents are training the clustering algorithm on more labelled instances, however this does not affect the performance of agents, as seen in Figure 3(b). This implies that the labels are not affecting the final clusterings, and ultimately whether or not a trustee's features can be associated with a cluster regardless of the label is an important aspect of CIM. Therefore, choosing a value for υ does not impact on CIM's performance. These results were evaluated with dynamic behaviour, and the same trend is true of static behaviour.



(a) Static Behaviour

(b) Dynamic Behaviour

(c) Legend

Fig. 1. Average utility per timestep.

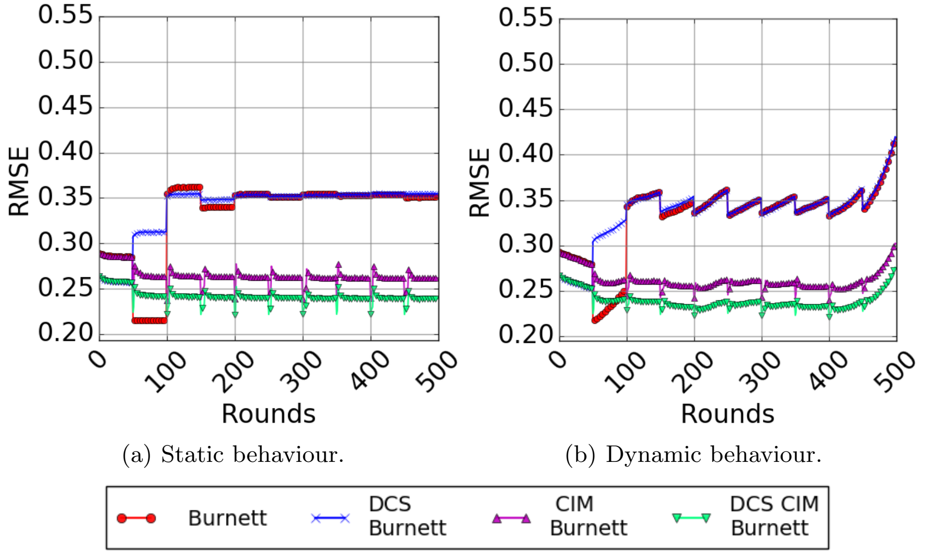
7 Conclusion

In this paper, we highlighted the importance of making agents aware of any class imbalance they might have in their history of past interactions, and the negative effects this has on stereotype assessment. CIM aims to improve agents' ability to assess interaction partners by adjusting their assessment technique depending on the class imbalance they have with respect to a potential partner. Additionally, CIM monitors for behaviour changes in agents, since this may also render data-driven stereotype methods ineffective. One advantage of this approach is that it overcomes the class imbalance problem without using exploration. Instead, agents are made aware of the imbalance and use this to inform their decision making.

Future work includes, propagating DCS assessments to improve initial assessments, as currently DCS only offers insight about two agents who appear very similar. However, agents would have to consider who they trust to accept a DCS evaluation from. We would also like to use a semi-supervised, density based, online clustering technique as a stereotype model because using the same motivation as for CIM, a semi-supervised approach could improve accuracy and an online clustering algorithm is efficient enough to be updated with every new instance rather than in intervals as in previous work. This would also allow CIM (with or without DCS) to work independently of any other stereotype model.

References

1. Burnett, C., Norman, T.J., Sycara, K.: Bootstrapping trust evaluations through stereotypes. In: Proceedings of the 9th International Conference on Autonomous



(a) Static behaviour.

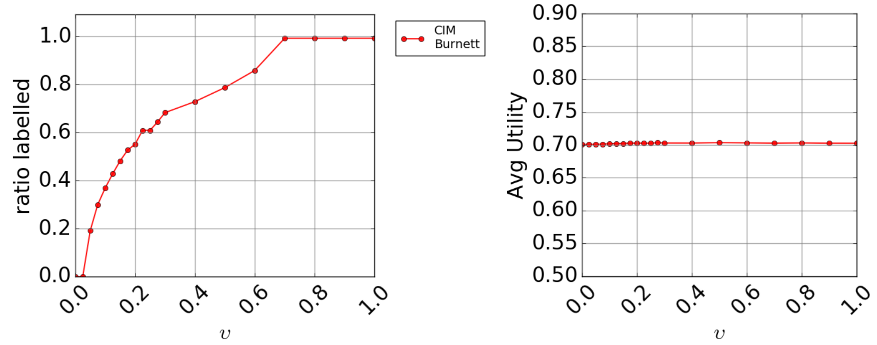
(b) Dynamic behaviour.

(c) Legend

Fig. 2. RMSE of *a priori* per timestep.

Agents and Multiagent Systems. pp. 241–248 (2010)

2. Burnett, C., Norman, T.J., Sycara, K.: Stereotypical trust and bias in dynamic multiagent systems. ACM Transactions on Intelligent Systems and Technology **4**(2), 26 (2013)
3. Chawla, N.V.: Data mining for imbalanced datasets: An overview,. In: Data mining and knowledge discovery handbook, pp. 875–886. Springer (2009)
4. Chawla, N.V., Japkowicz, N., Kotcz, A.: Special issue on learning from imbalanced data sets. ACM Sigkdd Exploration Newsletter **6**(1), 1–6 (2004)
5. Debra, M., Weick, K.E., Kramer, R.M.: Swift trust and temporary groups. Trust in organizations: Frontiers of theory and research p. 166 (1995)
6. Fadul, J., Hopkinson, K., Sheffield, C., Moore, J., Andel, T.: Trust management and security in the future communication-based" smart" electric power grid. In: System Sciences (HICSS), 2011 44th Hawaii International Conference on. pp. 1–10. IEEE (2011)
7. Gambetta, D.: Can we trust trust? Trust: Making and breaking coooperative relations **13**, 213–237 (2000)
8. Gray, E., Jensen, C., O'Connell, P., Weber, S., Seigneur, J.M., Chen, Y.: Trust evolution policies for security in collaborative ad hoc applications. Electronic Notes in Theoretical Computer Science **157**(3), 95–111 (2006)
9. Huynh, T.D., Jennings, N.: 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. vol. 110, p. 18 (2004)
10. Jøsang, A., Ismail, R.: The beta reputation system. Proceedings of the 15th Bled Electronic Commerce Conference **5**, 2502–2511 (2002)



(a) Ratio of labelled to unlabelled instances.

(b) Average Utility per Agent per Timestep for different v.

Fig. 3. Effects of v.

11. Liu, X., Datta, A., Rzadca, K.: Trust beyond reputation: A computational trust model based on stereotypes. Electronic Commerce Research and Applications **12**(1), 24–39 (2013)
12. Player, C., Griffiths, N.: Addressing concept drift in reputation assessment. In: Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems. pp. 2048–2050 (2018)
13. Regan, K., Poupart, P., Cohen, R.: Bayesian reputation modelling in e-marketplaces sensitive to subjectivity, deception and change. In: Proceedings of the National Conference on Artificial Intelligence (2006)
14. Resnick, P., Kuwabara, K., Zeckhauser, R., Friedman, E.: Reputation systems. Communications of the ACM **43**(12) (2000)
15. Riolo, R.L., Cohen, M., Axelrod, R.: Evolution of cooperation without reciprocity. Nature **414**(6862), 441–443 (2001)
16. Ruiz, C., Menasalvas, E., Spiliopoulou, M.: C-denstream: Using domain knowledge on a data stream. In: International Conference on Discovery Science. pp. 287–301. Springer (2009)
17. Sensoy, M., Yilmaz, B., Norman, T.J.: Stage: Stereotypical trust assessment through graph extraction. Computational Intelligence **32**(1), 72–101 (2016)
18. Sicari, S., Rizzardi, A., Grieco, L.A., Coen-Porsini, A.: Security, privacy and trust in internet of thigns: The road ahead. Computer Networks **76**, 146–164 (2015)
19. Teacy, L., Luck, M., Rogers, A., Jennings, N.: An efficient and versatile approach to trust and reputation using hierarchical bayesian modelling. Artificial Intelligence **193**, 149–185 (2012)