Addressing Concept Drift in Reputation Assessment

Caroline Player
University of Warwick
Coventry, West Midlands
c.e.player@warwick.ac.uk

Nathan Griffiths
University of Warwick
Coventry, West Midlands
nathan.griffiths@warwick.ac.uk

ABSTRACT
Evaluating trust and reputation in environments where agent behaviours are highly dynamic is challenging. In this paper, we address the limitations of existing methods when agent behaviours can change at varying speeds and times across a system. Modelling such environments as a multi-agent system, we propose a method that expands on existing trust, reputation and stereotype models and uses concept drift detection to identify and exclude unrepresentative past experiences. Our method improves the selection of relevant data for evaluating trust and reputation, without excluding relevant historical data (as is the case with sliding windows and forgetting factors). We show that our approach enables agents to achieve higher utility than existing methods, by more accurately and robustly evaluating trust and reputation with respect to dynamic agent behaviours.

1 INTRODUCTION
A service oriented architecture (SOA) is a system where service providers and consumers interact to achieve goals. With the growth of the Internet, SOAs are becoming more prevalent and can be used for mobile ad-hoc networks, crowd sourcing applications, the Internet of Things and peer-to-peer networks. However, this means exposing more devices to unreliable and potentially malicious interaction partners [28]. Therefore, it is vital to identify trustworthy interaction partners, but this is challenging in an open and dynamic environment. An SOA is frequently modelled using autonomous multi-agent systems (MAS) where agents use trust and reputation models to choose who to exchange information with [21, 25]. In MAS, trust is the expectation one agent has in another to perform a task, and is typically calculated using past experiences with the agent [7]. This is challenging when agents can frequently change their behaviour, because past experience is no longer an indication of future performance. We believe that it is realistic to assume that agents’ abilities may change at different times and speeds from others because their current quality of service (QoS) can be affected by time sensitive features including connectivity, task load and demand [17]. However, existing trust and reputation models commonly use sliding windows and forgetting factors, which are limited in their ability to select data which is representative of an agent’s current behaviour.

In an open network, where agents are continuously arriving and leaving, there may be agents that have not had any interactions and therefore there are no direct experiences from which to assess trust. However, agents may display observable features which are similar to the non-functional requirements of entities in an SOA which contribute to QoS. Where agents have observable features we can use stereotypes, an approach that assumes an agent might behave similarly to others who share the same observable features. One advantage of assuming agents have stereotypical behaviour is that the method to select relevant data is more proactive. For example, by identifying a negative change in an agent’s behaviour who shares a location with another agent, we might assume that connectivity there is currently poor and therefore neither agent is trustworthy. By identifying stereotypes we can also harness information from multiple agents to identify behaviour changes, for example, we will notice bad connectivity in a location by considering the experiences from all the agents in a location.

In this paper, we propose a method, the AdWin Tree, for agents to ignore data that is no longer relevant to trust and reputation assessment. This method integrates concept drift detection from the Adaptive Windowing (AdWin) algorithm with trust, reputation and stereotype assessments which are more suited to addressing the uncertainty in MAS. We evaluate our work using Beta Reputation System (BRS) and Burnett et al.’s stereotype decision tree [2, 12].

Our results show that the integration of change detection allows agents to perform better and more consistently when gradual and sudden drift in behaviour occurs for multiple agents. Our method is robust against noisy observable features, and we demonstrate that it outperforms existing work because the adaptive window does not require prior knowledge of the rates and times of behaviour change in order to tune parameters.

2 BACKGROUND
There are a number of existing trust, reputation and stereotype techniques that account for changes in agent behaviour. We discuss these models, their limitations, and introduce concept drift detection in this section.

2.1 Trust, Reputation and Stereotypes
In MAS, agents use their direct experiences with others to estimate how they will behave in the future [13]. There are a variety of models for estimating trust, reputation and stereotypical trust. Selecting the most appropriate depends on what information is available, including representations of outcomes, task decomposition, and observable features of agents. Additionally, the selection depends on whether agents know enough about the environment either in advance, or at run time, to accurately tune the parameters of the models. In this paper, we propose a method that avoids the requirement of knowing the rate of behaviour change, and times of any sudden change, in advance. Multi-dimensional approaches to trust can be appropriate if the necessary granularity of information is available. For example, agents using REGRET can rate different aspects of an interaction such as the price and quality of a product [22]. Some trust and reputation systems allow agents to have subjective opinions, for example an agent can be perceived as “good” or “quite
helpful”. These opinions can be used in fuzzy logic models, which take these subjective descriptions into account [8, 9, 16, 18, 24].

A reputation system collects, aggregates and distributes feedback about agents’ past behaviour [20]. In this work, we adopt Beta Reputation System (BRS), a mathematically rigorous reputation model which aggregates witness reports to estimate the expected behaviour of an agent and uses subjective logic to account for uncertainty when there is a small data sample [12]. BRS is general enough to suit many applications, however, any trust assessment method can be substituted, as described in Section 4. Extensions to BRS have been proposed to tackle issues such as bias in witness reports from lying reputation sources, or from witnesses with different perceptions of “good” and “bad” outcomes, such as the TRAVOS, BLADE and FIRE models, amongst others [11, 19, 26, 30]. However, there are no existing reputations systems that focus on statistically identifying whether witness reports are representative of the target agent’s current behaviour given that it may have changed. HABIT is a statistical reputation assessment technique which relaxes the requirement of knowing in advance the number of features that indicate QoS in an SOA [25]. However, there is limited consideration of the effect of time on trust assessment.

Inspired by Swift Trust [5], stereotypes estimate the behaviour of another agent by exploiting the intuition that an agent will behave similarly to those who have the same observable features. Our work focuses on environments in which agents’ behaviours can change, and if we assume that agents who share certain features will behave similarly then this implies that their behaviour will change in the same way. Stereotypes enable us to use interaction records from multiple agents to statistically monitor their behaviour over time and identify changes in an individual agent who is a member of that stereotype.

Stereotypes associate reputation with observable features (instead of agent identities as in traditional trust and reputation approaches). Trust can be calculated after an interaction, for example with BRS, and is recorded alongside the agent’s observable features. Burnett et al. use an M5 decision tree to identify correlations between observable features and trust [2–4]. The advantage of this approach is that the identified stereotypes are described by the leaves of the decision tree. The StereoTrust model assigns agents an a priori assessment of their behaviour based on the trust they have in other agents who are observably similar, weighted by the extent of that similarity [15]. STAGE describes agents using a graph-based ontology and identifies patterns of trustworthy and non-trustworthy agents [23]. This addresses the limitations of some machine learning techniques that can only find correlations between feature values and behaviours by finding patterns in the relationships between features.

Almost all stereotype methods assume that agents can fully and accurately perceive feature values, with the exception of Burnett et al. who use imputation to replace missing values [4]. In this paper, we also assume that agents can accurately perceive all of another agent’s observable features, as we focus on the problem of changing agent behaviours. We do not assume that agents can distinguish between the observable features that affect behaviour and those which do not and this is one of the challenges that we explore in our evaluation.

Some trust, reputation and stereotype models account for changes in behaviour over time with a sliding window or forgetting factor [11, 12, 14, 19, 26]. A sliding window of size n keeps the most recent n experiences. The intuition is that the most recent n instances capture an agent’s current behaviour, and older interaction records are no longer relevant. A forgetting factor has a similar effect, by retaining all records but weighting recent interactions higher than older ones. These approaches can be effective in simple environments, however several issues impact on their generality. First, choosing the window size or rate of forgetting is problematic. If an agent forgets old instances too quickly it will lose relevant data that could help make more accurate calculations. Conversely, if too much information is retained then agents will make assessments based on data that no longer represents current behaviours. The optimal values will depend on the application. Second, the optimal values for window sizes or forgetting factors can vary over time. Third, agents may not change their behaviour at the same rates, and so a global window or forgetting factor will not suit all agents. Finally, sliding windows and forgetting factors are only effective in coping with gradual change rather than sudden changes, which may occur at different times for different agents.

2.2 Concept Drift

We investigate whether concept drift techniques, which detect changes in correlations between a feature set, X, and a class value Y, can be applied to identifying changes in agent behaviour. Many machine learning models rely on the assumption that, aside from noise, the joint probability of the features and the target remains static over time i.e $P(X|Y) = P(X|Y)$ for all times t and t’ [29]. This assumption may hold for some offline learning tasks, but many applications have vast quantities of data that are continuously requiring online learning to adapt to changes in correlations. Concept drift techniques aim to detect when such relationships change over time, meaning $P_t(X|Y) = P_{t'}(X|Y)$ [27]. For example, when the connectivity in a location changes, that location may no longer correlate with the QoS originally learned from data in that area.

A concept can change gradually or suddenly. If the correlation between a set of feature values and a class value instantly changes to a new class value then sudden drift has occurred. A slow progression of one concept becoming less prevalent while another begins to dominate is known as gradual drift. Gradual drift is harder to detect, especially when the differences in the two concepts are not empirically large but are important [10].

The change detection element of concept drift algorithms identifies a point in time where concept drift has occurred. This can be used to update interaction data records for trust, reputation and stereotype assessment. In concept drift literature there are two common techniques to manage the data once change has been detected. First, to forget data according to a time based function, and second to use a window of recent data of fixed or variable size [6]. These share many of the problems described above when they are used with trust and reputation techniques. However, some concept drift techniques can handle more complex scenarios such as detecting and handling varying rates of change.

The Adaptive Windowing (AdWin) model uses a variable window size that can adapt to sudden and gradual changes by saving
data that is statistically assessed to be drawn from the same distribution as previous data and therefore is relevant [1]. If a change is detected then any data from before the change is forgotten, which is synonymous with shrinking the window. The method we propose in this paper integrates concept drift detection using AdWin to a trust, reputation and stereotype assessment system.

3 AGENT MODEL

This section describes the agent interaction environment which is based on the approaches from Burnett et al. and BRS [2, 12].

A set of agents is divided into trustors, \( \mathcal{A}_{tr} \), and trustees, \( \mathcal{A}_{te} \). The two sets are connected in a complete bipartite graph where a trustor represents a service consumer, \( i \), trying to identify a trustworthy service provider, a trustee, \( j \). Agents can be described by their ID and their observable features, denoted as a vector, \( \vec{f}_j \), for agent \( j \). Each feature corresponds to a characteristic of an agent, for example the technical specifications of a device in a network. Some features correlate with behaviour, known as relevant features. Additionally, agents have noisy features, also known as irrelevant features, whose values are assigned at random to the agent. This represents a realistic assumption that not all observable characteristics of an agent affect, or are indicative of, their behaviour. Such features are included because we cannot assume that agents know which features correlate with behaviour in advance. The number of relevant and noisy features each agent has, \( n_{rf} \) and \( n_{nf} \) respectively, are experimental parameters.

We define \( k \) profiles which dictate the relevant features, behaviour and rates of behaviour changes for the agents of each profile. This enforces the assumption that agents who share relevant feature values behave the same. Noisy feature values are assigned randomly, independently of the profile, and will differ between agents of the same profile. Relevant feature values of the profiles are the centroids generated from applying \( k \)-means clustering to all permutations of \( n_{rf} \) relevant features. Table 1 presents example profiles where \( k = 5 \) and the values are taken from one of the experimental runs. We use binary feature values for simplicity and comparison to existing work, however the stereotype model we use can handle numeric and categorical attributes. Table 1 depicts the noisy features appended to the end of the feature vector for illustration, however, they may be interleaved. Trustors do not

<table>
<thead>
<tr>
<th>Profile</th>
<th>Initial Behaviour</th>
<th>( p^{0}_{Gr} )</th>
<th>( p^{0}_{Su} )</th>
<th>( \vec{f}_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.833</td>
<td>0.073</td>
<td>0.001</td>
<td>1,0,1,0,1,0,</td>
</tr>
<tr>
<td>1</td>
<td>0.5</td>
<td>0.000</td>
<td>0.002</td>
<td>0,0,1,0,1,1,1,0</td>
</tr>
<tr>
<td>2</td>
<td>0.167</td>
<td>0.097</td>
<td>0.000</td>
<td>0,1,0,0,1,1,0,1</td>
</tr>
<tr>
<td>3</td>
<td>0.667</td>
<td>0.092</td>
<td>0.005</td>
<td>0,1,0,0,1,0,1,1</td>
</tr>
<tr>
<td>4</td>
<td>0.333</td>
<td>0.023</td>
<td>0.003</td>
<td>1,1,1,0,1,1,0,0</td>
</tr>
</tbody>
</table>

The behaviour of agent \( i \), defined by the profile \( i \) belongs to, is represented as a real value \( B_i \in [0, 1] \) indicating the proportion of interactions they behave well in. For example, if \( B_i = 0.7 \) then agent \( i \) will behave and be perceived as good in 7 out of 10 interactions on average. Trust, reputation and stereotype systems aim to estimate the behaviour of agents, in order to identify the agent most likely to behave well in its next interaction [12]. The value of initial behaviours are evenly distributed within the range \([0, 1]\) but can then change over time gradually or suddenly. We define the environmental parameters \( p_{Gr} \) and \( p_{Su} \) such that each profile \( n \) is assigned a local random value, \( p^{n}_{Gr} \) in the range \([0, p_{Gr}]\), and \( p^{n}_{Su} \) in the range \([0, p_{Su}]\). This allows profiles to change behaviour at different speeds. Gradual change occurs by assigning each profile a random target behaviour and probabilistically with a chance of \( p^{n}_Gr \), altering their current behaviour by 0.01 towards the target. Similarly for sudden change, a random target behaviour is chosen and immediately switched to with a probability of \( p^{n}_Su \) each round. Once a profile’s actual behaviour has reached the target behaviour, a new random target behaviour is chosen for that profile. Figure 1 depicts how the value for behaviours of each profile change over time in one experimental run given the parameters from Table 1.

One timestep, in which the trustees choose one trustee to interact with who they believe has the highest expected utility, is considered a round. Trustees can only be chosen by one trustee per round and the order in which trustees choose their partners is random each round. An interaction in an SOA represents an exchange of information between a service provider and consumer. According to BRS, the trustee perceives the outcome, determined by the trustee’s behaviour, as either good or bad. Trustees maintain a history of their interactions, \( O \), as a set of tuples in the form \((j, \vec{f}_j, t, r_j, s_j)\), where \( j \) is the trustee’s ID, \( \vec{f}_j \) is \( j \)’s feature vector, \( t \) is the time of the interaction, \( r_j \) and \( s_j \) are the number of good and bad experiences with \( j \) respectively up to time \( t \), including the witness reports \( i \) received at the time of the interaction.

As we model open systems, at the end of a round trustees can leave the network with a probability \( P_{te} \). They are replaced by a new agent of the same profile, thus maintaining a constant distribution of agent behaviours.
4 ESTIMATING EXPECTED BEHAVIOUR

Our method is evaluated using BRS and Burnett’s decision tree and we will refer to this instantiation of our method as the AdWin Tree, to identify relevant data for calculating expected behaviour [2, 12]. We use Burnett’s stereotype model because it outputs a set of identified stereotypes, \( S \), which we can monitor. These stereotypes are found by correlating the observable features with an agent’s trust which is calculated using a trust and reputation function \( TR \). Our method takes in the set of discovered stereotypes, the existing data, and the new outcome after an interaction at time \( t \), \( o^t \), and if a change is detected, it will alter the dataset, i.e \( O \leftarrow f(S, O, o^t) \). The trust and reputation model used, \( TR \), could be substituted, provided that its output is normalised to be in the range \([0, 1]\). Choosing an appropriate \( TR \) is application dependent.

4.1 Bootstrapping Trust with Stereotypes

Recall that trustor \( i \) maintains an interaction history \( O \), where an entry has the number of good and bad interactions with agent \( j \), \( r_j \) and \( s_j \) respectively, up to time \( t \). For now, we assume the tuple maintains relevant data. The belief, \( b \), in a trustee is their expected behaviour based solely on their previous interactions, calculated using Equation 1. To account for uncertainty in this belief, an a priori, \( a \), is weighted by a factor, \( u \), calculated using Equation 2, which decreases as more interaction experiences are collected, signifying a higher confidence in the belief factor. The default value for \( a \) is 0.5, which assumes that an agent is neither good nor bad. Once the trustor has enough data to build the stereotype model, \( a \) is replaced by stereotypical trust. The overall expected behaviour, \( trust(j) \), is calculated using subjective logic using Equation 4 according to BRS [12].

\[
b_j = \frac{r_j}{r_j + s_j + 2} \quad (1)
\]

\[
u_j = \frac{2}{r_j + s_j + 2} \quad (2)
\]

\[a_j = \begin{cases} \text{stereotype}(\tau_j), & \text{if using stereotypes} \\ 0.5, & \text{otherwise} \end{cases} \quad (3)
\]

\[trust(j) = b_j + u_j \times a_j \quad (4)
\]

The values of \( r_j \) and \( s_j \) are an aggregation of good and bad interactions with \( j \) from available trustors as well as personal interactions, as defined in Equation 5. We assume that witness reports are honest, and therefore both direct and indirect experiences are weighted equally [2, 12]. If we removed this assumption, then other trust and reputation models could substitute BRS to tackle biased and lying reports [19, 25, 26]. If the trustor has detected behaviour change in the trustee, then it will only request witness reports from tuples timestamped after that change.

\[r_j = \sum_{t\tau} r_j^t r, \quad s_j = \sum_{t\tau} s_j^t r \quad (5)
\]

The learning interval, \( L \), determines how frequently a trustor rebuilds the stereotype model and also the size of the fixed window when the adaptive window is not being used. The data used to train the decision tree comes from each record in the trustee’s interaction history, \( O \). The feature set and class attribute used are: \((\tau, trust(j))\), where \( trust(j) \) is calculated using \( r_j \) and \( s_j \) from the respective entry in \( O \) and Equation 4. An M5 decision tree can handle a numeric class attribute because each leaf has a linear regression model (LM), trained on the class values of the data at the respective leaf. To assess stereotypical trust, trustors classify the observable features of a trustee, which outputs the a priori expected behaviour in the range \([0, 1]\). When there are few direct experiences with the trustee the uncertainty will be high, giving more weight to stereotypical trust. The decision tree allows trustors to identify stereotypes without knowing how many agent profiles exist in advance.

By default, the M5 algorithm prunes the decision tree. This can prevent over-fitting behaviour assessments to irrelevant features and overestimating the number of stereotypes in the population. However, in the context of assessing behaviour in MAS, the class value is an estimate of agent behaviour rather than a true value, therefore it can be inaccurate. Due to this error, pruning the tree is sometimes counter productive. We modify the M5 algorithm to not prune a node if the subtree has the same error, where originally it would get pruned. This only prevents pruning occasionally, however, this preserves some identified stereotypes by accounting for a little more error.
4.2 Adaptive Window Decision Tree (AdWin Tree)

Each leaf of a trustor’s decision tree represents their estimate of an agent profile i.e a stereotype. We build an AdWin model at each leaf, as depicted in Figure 2, to monitor for profiles changing at different rates and times. The accuracy of the concept drift model is bottlenecked by how accurately the decision tree identifies agent profiles. If one leaf incorrectly represents multiple profiles the data will be more noisy and prevent the concept drift model from accurately detecting drift in one profile. We implement the AdWin 2 algorithm which uses an exponential histogram data structure to improve time and space efficiency over the original AdWin formulation [1]. The detected point of change can still be extracted to adjust the memory for trust and reputation models. However, we describe AdWin because its linear data structure allows for a more logical explanation of the algorithm.

4.2.1 Detecting Drift. Once the decision tree is built, an AdWin model is created at each leaf using the data at that leaf. To detect concept drift in the first L interactions, each instance must be added to the new concept drift model one at a time. Any drift detection is handled as described below in Section 4.2.2.

Once a tree is built with data where no drift is detected, the trustee proceeds with interactions. After each interaction, the binary outcome is appended to the window of the AdWin model at the appropriate decision tree leaf, which is found by classifying the trustee’s observable features. This binary outcome does not affect the LM model at that leaf because this is only refactored when the tree is rebuilt, which is every Ln interactions or when concept drift is detected. The AdWin model concatenates the new result to the end of a variable size window, which contains the data at that leaf up to that point in time. To detect if drift has occurred, the window is divided into every possible split of two subwindows. The distributions found to fit the data of each subwindow are compared and if they are within a margin of similarity then it is assumed that no concept drift has occurred.

The inputs to the AdWin algorithm are a confidence value \( \delta \in (0, 1) \) and a sequence of real values \( x_1, x_2, ..., x_t \in (0, 1) \) where the value of \( x_t \) is made available at time \( t \), and represents the outcomes from interactions. We use \( \delta = 0.2 \) as a suggested value from the original work, and from experimental results we saw little difference from changing this value in this application of the algorithm. Each \( x_t \) is generated according to a distribution \( D_t \), however the distribution may change over time. This represents the changing behaviour of agents in a profile. The mean, expected value of \( x_t \), and expected variance of \( x_t \) are \( \mu_t \) and \( \sigma^2_t \) according to \( D_t \), where \( \mu_t \) and \( \sigma^2_t \) are unknown for all \( t \) and can only be estimated by trustors assessing behaviour.

A window \( W \) holds the most recently read \( x_t \), with \( \hat{\mu}_W \) being the observed average of the elements in \( W \), and \( \mu_W \) is the unknown but true average of the elements in \( W \). The window can change size and we denote its length at any point as \( n \). The values of \( \mu_t \), the mean of \( D_t \) from which \( x_t \) was drawn at time \( t \), can oscillate over time as concept drift occurs. Therefore, it is possible that neither \( \mu_W \) or the observed mean \( \hat{\mu}_W \) will be near to \( \mu_t \), even as the length of the window gets increasingly long. However, \( \hat{\mu}_W \) will become a closer estimate of \( \mu_W \) as \( n \) increases.

Concept drift is identified when the window \( W \) is divided into two subwindows, which exhibit "distinct enough" averages, implying their respective expected values originate from different distributions. In this case, the older of the two subwindows is deleted from the original window \( W \). Therefore, the window decreases while the statistical tests show that \( \mu_t \) has changed significantly in \( W \) within the confidence \( \delta \). Another test could be used, but in this paper we choose the default test from the AdWin technique, as explained below. Each step of AdWin outputs the value \( \hat{\mu}_W \) as an approximation of \( \mu_W \).

The two subwindows are considered to be generated by different distributions if the difference between their means is larger than \( \varepsilon_{cut} \), a variable calculated with the user specified confidence input \( \delta \). For a partition of window \( W \) into \( W_0 \cdot W_1 \), \( \varepsilon_{cut} \) is computed as follows: \( n_0 \) and \( n_1 \) are the lengths of \( W_0 \) and \( W_1 \) respectively \( (n = n_0 + n_1) \). Let \( \bar{\mu}_W_0 \) and \( \bar{\mu}_W_1 \) be the averages of the values in their respective subwindows. And \( \mu_{W_0} \) and \( \mu_{W_1} \) are their expected values, resulting in the following:
with their behaviour. The average of a small number of interactions will not perfectly align because the AdWin Tree, to using fixed windows. Figure 3d is the legend for all the results in this section. Figures 3a, 3b and 3c show an increasing volatility of dynamic behaviours and how well agents perform best. However, we include some results on the error of our model to demonstrate that it improves upon existing work and this produces a correct ranking may have quite a high error but still be statistically significant with \( p \) < 0.001. Each profile \( n \) has a rate of gradual change in the range 0 to \( p_{Gr} \), and not all profiles are changing at the maximum rate \( p_{Gr} \). The values of \( p_{Gr} \) and \( p_{Su} \) have been chosen because a profile \( n \) which has a rate of gradual change in the range 0 to \( p_{Gr} \), and not all profiles are changing at the maximum rate \( p_{Gr} \). The values of \( p_{Gr} \) and \( p_{Su} \) have been chosen because a profile \( n \) whose \( p^*_{Gr} \) is the maximum value of \( p_{Gr} \), could have such volatile behaviour that the data received from interacting with agents of profile \( n \) appears as noise. Although the parameter values seem low, Figure 1 illustrates how quickly behaviours are changing when \( p_{Gr} = 0.1 \) and \( p_{Su} = 0.005 \).

If a fixed sliding window is the optimal size for the rate of gradual change of behaviour, then it can move with the incoming data such that it retains only representative data. A disadvantage of the adaptive window is it must collect sufficient unrepresentative data before it can detect that there has been a change in the underlying distribution generating the outcomes. Therefore, before the change can be detected, but after it has happened, the adaptive window will be making predictions with some unrepresentative data. This is using either the AdWin Tree or fixed windows of sizes of 50, 100 or 200, perform. The adaptive window outperforms the fixed window sizes consistently. We can also observe that no fixed window size is better than another. Different profiles are subject to different rates of behaviour change, and therefore each profile would suit a different window size. However, the adaptive window can change for each profile, and also does not need to know the rate of change in advance. Using a fixed window would require this information in advance to apply the best window size.

5.2 Gradual and Sudden Drift

Sliding windows are designed to handle a known, constant rate of gradual drift, the results of which are shown in Figure 4, where Figures 4a and 4b show different maximum rates of gradual change, \( p_{Gr} \). Each profile \( n \) has a rate of gradual change in the range 0 to \( p_{Gr} \), and not all profiles are changing at the maximum rate \( p_{Gr} \). The values of \( p_{Gr} \) and \( p_{Su} \) have been chosen because a profile \( n \) whose \( p^*_{Gr} \) is the maximum value of \( p_{Gr} \), could have such volatile behaviour that the data received from interacting with agents of profile \( n \) appears as noise. Although the parameter values seem low, Figure 1 illustrates how quickly behaviours are changing when \( p_{Gr} = 0.1 \) and \( p_{Su} = 0.005 \).

If a fixed sliding window is the optimal size for the rate of gradual change of behaviour, then it can move with the incoming data such that it retains only representative data. A disadvantage of the adaptive window is it must collect sufficient unrepresentative data before it can detect that there has been a change in the underlying distribution generating the outcomes. Therefore, before the change can be detected, but after it has happened, the adaptive window will be making predictions with some unrepresentative data. This is using either the AdWin Tree or fixed windows of sizes of 50, 100 or 200, perform. The adaptive window outperforms the fixed window sizes consistently. We can also observe that no fixed window size is better than another. Different profiles are subject to different rates of behaviour change, and therefore each profile would suit a different window size. However, the adaptive window can change for each profile, and also does not need to know the rate of change in advance. Using a fixed window would require this information in advance to apply the best window size.
We briefly demonstrate that both models are robust to noisy features especially, this means it cannot accurately detect changes in behaviour which occur specifically in one profile. This allows direct experiences to take precedence according to Equation 4, and these records are more up to date compared to the interaction records with agents who have left the system already but whose data may still be contributing to stereotypical trust assessment.

Sliding windows do not account for sudden drifts, however these may occur in many applications. Figure 5 isolates the effect of sudden drift i.e $p_{GR} = 0$. An agent profile changing its behaviour to a new random value is uncommon, therefore low values for $p_{SR}$ are considered, but the difference in sudden drift between Figures 5a and 5b is large enough that the results fluctuate more for both models. As the sudden changes in behaviour become more common both models perform slightly worse, but the AdWin Tree remains robust against this volatility. The AdWin Tree recovers quickly by shrinking the window when sudden change is detected for one profile, while the original model takes longer to forget old behaviour because of its fixed window size.

Sudden changes in behaviour may occur after any amount of time of steady behaviour, and therefore the AdWin Tree benefits not just from its ability to shrink the window when a change in behaviour is detected, but also from growing the window when no change is detected. As the window size increases there is significantly more data with which to make a more accurate trust assessment compared to the limited size of the fixed window.

5.3 Noisy Features

We briefly demonstrate that both models are robust to noisy features in Figure 6. Identifying noisy features when behaviour is static is an important but relatively easy task compared to when there is concept drift. As behaviours change, some noisy features may coincidentally correlate with behaviour for a period before behaviour changes again, making them harder to distinguish. Ideally the model should handle a high proportion of noisy features, as we do not assume that agents have advance knowledge about which features correlate with behaviour. All previous results have been demonstrated where $n_{f} = 6$ and $n_{nf} = 6$.

One major bottleneck to the efficacy of this work is how accurately the agent profiles are recognised as stereotypes. Each leaf of the decision tree represents a stereotype, where a stereotype is an estimation of a profile. We can see from Figure 7 that neither model has 5 leaves to represent the 5 profiles in the system. While the stereotype tree uses a Linear Regression model for predicting trust values at its leaves and therefore has this final ability to distinguish some different trust values for different profiles, the data stored from experiences are all at this one leaf. For the adaptive window especially, this means it cannot accurately detect changes in behaviour which occur specifically in one profile.

### Figure 6: Varying noisy features

Both models are fairly robust to noisy features, in part because a trust and reputation algorithm does not take noisy features into account, and trustors give precedence to direct experiences over stereotypes when they are available. However, more accurately identifying stereotypes would improve this further, especially in contexts where there is a limited availability of direct experiences.

5.4 Error

Selecting the partner with the highest chance of performing well requires some level of accurate trust assessment. Here, we show the root mean squared error (RMSE) of the AdWin Tree compared to the fixed window model is considerably lower, and this will be a contributing factor to its performance. Initially, the original model has relatively low error, however as behaviour begins to change, when the stereotype model is rebuilt, it has conflicting data compared to the adaptive model which is selecting only data it believes reflects current behaviours.

Ultimately, only the correct ranking of a few top agents is required to select the best agent, even if the assessments themselves have a high level of error. Equally, a technique with low error may not have identified the true best behaving agent, preventing it from performing well. As discussed earlier, one reason for an error rate as high as 0.2 is because the profiles may not have been accurately identified, and therefore stereotype predictions for an agent are derived from data from a mixture of profiles.
We demonstrated that compared to the most prominent approach when agents’ behaviours may be subject to different rates of change, individual agents rather than stereotypes, as this would be particularly helpful in identifying cheating agents who are acting maliciously but display the observable features of a trustworthy partner.

6 DISCUSSION AND CONCLUSION

Changing agent behaviours can prevent existing trust, reputation and stereotype models from accurately assessing agents because they rely on past interaction data which may no longer represent their current behaviour. Existing work uses methods which give a higher weighting to more recent interactions. However, these techniques use the same weighting for all agents’ past experiences when agents’ behaviours may be subject to different rates of change or to sudden changes.

In this paper, we proposed the AdWin Tree method and demonstrated how agents using concept drift detection can select only relevant data for each agent. Basing this selection on drift detection rather than time improves their partner selection choices when agents were subject to a variety of gradual and sudden behaviour changes. Our work can extend any existing stereotype model to allow agents to detect the rates of change in an environment, as these will vary depending on the application.

The AdWin Tree is resilient against changes in agent behaviour, performing consistently, especially when there are sudden changes. We demonstrated that compared to the most prominent approach currently used in trust and reputation, a sliding fixed window, our model performs either the same or better, with the advantage of not requiring any parameters to be set. As the window can not only shrink when changes are detected, but can also grow while behaviour is believed to be static, the predictions can be far more accurate than using a window of fixed size. The model is still robust against a high proportion of noisy observable agent features and it has a lower RMSE for estimating agent behaviours.

The performance of the adaptive model is bottlenecked by how accurately the decision tree can identify the agent profiles as stereotypes. Each AdWin model is applied to an identified stereotype, and if these are inaccurately identified then multiple profiles will be analysed by one AdWin model, making it harder to identify changes in just one profile.

Future work includes investigating contexts where agents are subject to virtual as well as real concept drift. Another direction is to explore monitoring behaviour changes or differences in individual agents rather than stereotypes, as this would be particularly helpful in identifying cheating agents who are acting maliciously but display the observable features of a trustworthy partner.

ACKNOWLEDGMENTS

The lead author gratefully acknowledges funding from the UK EPSRC (grant no. EP/L016400/1), the Centre for Doctoral Training in Urban Science.

REFERENCES


