Are You Doing What I Think You Are Doing? Criticising Uncertain Agent Models

Are You Doing What I Think You Are Doing? Criticising Uncertain Agent Models

Stefano V. Albrecht
School of Informatics
University of Edinburgh
Edinburgh EH8 9AB, UK
s.v.albrecht@sms.ed.ac.uk

Subramanian Ramamoorthy
School of Informatics
University of Edinburgh
Edinburgh EH8 9AB, UK
s.ramamoorthy@ed.ac.uk

Abstract

The key for effective interaction in many multiagent applications is to reason explicitly about the behaviour of other agents, in the form of a *hypothesised* behaviour. While there exist several methods for the construction of a behavioural hypothesis, there is currently no universal theory which would allow an agent to contemplate the correctness of a hypothesis. In this work, we present a novel algorithm which decides this question in the form of a frequentist hypothesis test. The algorithm allows for multiple metrics in the construction of the test statistic and learns its distribution during the interaction process, with asymptotic correctness guarantees. We present results from a comprehensive set of experiments, demonstrating that the algorithm achieves high accuracy and scalability at low computational costs.

1 INTRODUCTION

A common difficulty in many multiagent systems is the fact that the behaviour of other agents may be initially unknown. Important examples include adaptive user interfaces, robotic elderly assistance, and electronic markets. Often, the key for effective interaction in such systems is to reason explicitly about the behaviour of other agents, typically in the form of a *hypothesised* behaviour which makes predictions about future actions based on a given interaction history.

A number of methods have been studied for the construction of behavioural hypotheses. One method is to use opponent modelling techniques to learn a behaviour from the interaction history. Two well-known examples include fictitious play (Brown, 1951) and case-based reasoning (Gilboa and Schmeidler, 2001), as well as their many variants. Another method is to maintain a set of possible action policies, called types, over which a posterior belief is computed based on the interaction history (Albrecht and Ramamoorthy, 2014; Gmytrasiewicz and Doshi, 2005). The hypothesis is then obtained by using the posterior to mix the types. Related methods have been studied in the plan recognition literature (Carberry, 2001; Charniak and Goldman, 1993).

The learned behaviours (or models) of these methods can be viewed as hypotheses because they are eventually either true or false (subject to the various assumptions they are based on), and because they are *testable*. Thus, the following is a natural question: given an interaction history H and a hypothesis π^* for the behaviour of an agent, does the agent indeed behave according to π^*? There are several ways in which an answer to this question could be utilised. For instance, if we persistently reject the hypothesis π^*, we may construct an alternative hypothesis or resort to some default plan of action (such as a "maximin" strategy).

Unfortunately, the above methods for hypothesis construction do not provide an answer to this question. Some opponent modelling methods use goodness-of-fit measures (e.g. those that rely on maximum likelihood estimation), but these measures describe how well the model fits the data (i.e. interaction history) and not necessarily if the model is correct. Similarly, the posterior belief in the type-based approach quantifies the relative likelihood of types (relative to a set of alternative types) but not the *correctness* of types.

To illustrate the source of difficulty, consider the below excerpt of an interaction process between two agents which can choose from three actions. The columns show, respectively, the current time t of the interaction, the actions chosen by the agents at time t, and agent 1's hypothesised probabilities with which agent 2 would choose its actions at time t, based on the prior interaction history.

t	(a_1^t, a_2^t)	π_2^*
1	$(1, 2)$	$\langle .3, .1, .6 \rangle$
2	$(3, 1)$	$\langle .2, .3, .5 \rangle$
3	$(2, 3)$	$\langle .7, .1, .2 \rangle$
4	$(2, 3)$	$\langle .0, .4, .6 \rangle$
5	$(1, 2)$	$\langle .4, .2, .4 \rangle$

Assuming that the process continues in this fashion, and without any restrictions on the behaviour of agent 2, how

should agent 1 decide whether or not to reject its hypothesis about the behaviour of agent 2?

A natural way to address this question is to compute some kind of *score* from the information given in the above table, and to compare this score with some manually chosen rejecting threshold. A prominent example of such a score is the empirical frequency distribution (Conitzer and Sandholm, 2007; Foster and Young, 2003). While the simplicity of this method is appealing, there are two significant problems: (1) it is far from trivial to devise a scoring scheme that reliably quantifies "correctness" of hypotheses (for instance, an empirical frequency distribution taken over all past actions would be insufficient in the above example since the hypothesised action distributions are changing), and (2) it is unclear how one should choose the threshold parameter for any given scoring scheme.

In this work, we present an efficient algorithm which decides this question in the form of a frequentist hypothesis test. The algorithm addresses (1) by allowing for multiple scoring criteria in the construction of the test statistic, with the intent of obtaining an overall more reliable scoring scheme. The distribution of the test statistic is then learned during the interaction process, and we show that the learning is asymptotically correct. Finally, analogous to standard frequentist testing, the hypothesis is rejected at a given point in time if the resulting p-value is below some "significance level". This eliminates (2) by providing a uniform semantic for rejection that is invariant to the employed scoring scheme. We present a comprehensive set of experiments, demonstrating that our algorithm achieves high accuracy and scalability at low computational costs.

Of course, there is a long-standing debate on the role of statistical hypothesis tests and quantities such as p-values (e.g. Gelman and Shalizi, 2013; Berger and Sellke, 1987; Cox, 1977). The usual consensus is that p-values should be combined with other forms of evidence to reach a final conclusion (Fisher, 1935), and this is the view we adopt as well. In this sense, our method may be used as part of a larger machinery to decide the truth of a hypothesis.

2 RELATED WORK

In addition to the related works mentioned in the previous section, there are a number of other related research areas:

There exists a large body of literature on what is often referred to as *model criticism* (e.g. Bayarri and Berger, 2000; Meng, 1994; Rubin, 1984; Box, 1980). Model criticism attempts to answer the following question: given a data set D and model M, could D have been generated by M? This is analogous to our question, in which D is a sequence of observed actions of some agent and M is a hypothesised behaviour for that agent. However, in contrast to our work, model criticism usually assumes that the data are independent and identically distributed, which is not the case in the interactive settings we consider.

A related problem, sometimes referred to as *identity testing*, is to test if a given sequence of data was generated by some given stochastic process (Ryabko and Ryabko, 2008; Basawa and Scott, 1977). Instead of independent and identical distributions, this line of work assumes other properties such as stationarity and ergodicity. Unfortunately, these assumptions are also unlikely in interaction processes, and the proposed solutions are very costly.

Model criticism and identity testing are not to be confused with *model selection*, in which two or more alternative models are under consideration (e.g. Vehtari and Ojanen, 2012). Similarly, we do not consider alternative hypotheses. However, our method can be applied individually to multiple hypotheses, or the hypotheses may be fused into a single hypothesis using a posterior belief (Albrecht and Ramamoorthy, 2014; Gmytrasiewicz and Doshi, 2005).

Another related problem is that of *model checking*, which attempts to verify that a given system (or model) satisfies certain formal properties (Clarke et al., 1999). Recently, Albrecht and Ramamoorthy (2014) applied the concept of probabilistic bisimulation (Larsen and Skou, 1991) to the question of "incorrect" hypotheses and showed that a certain form of optimality is preserved if a bisimulation relation exists. However, their work is not concerned with establishing whether or not a given behavioural hypothesis is correct, and their analysis is performed *before* any interaction.

Our method can be viewed as *passive* in the sense that it does not actively probe different aspects of the hypothesis, and we show in Section 5 that this can be a drawback. This is in contrast to methods such as (Carmel and Markovitch, 1999), which promote active exploration. However, this exploration comes at high computational costs and limits the structure of hypotheses, such as deterministic finite state machines. On the other hand, our method has low computational costs and leaves the structure of the hypothesis open.

3 PRELIMINARIES

We consider a sequential interaction process with m agents. The process begins at time $t = 0$. At each time t, each agent $i \in \{1, ..., m\}$ receives a signal s_i^t and chooses an action a_i^t from a finite set of actions A_i. (Agents choose actions simultaneously.) The process continues in this fashion indefinitely or until some termination criterion is satisfied.

The signal s_i^t specifies information that agent i receives at time t and may in general be the result of a random variable over past actions and signals. For example, s_i^t may be a discrete system state and its dynamics may be described by some stochastic transition function. Note that we allow for asymmetric information (i.e. $s_i^t \neq s_j^t$). For example, s_i^t may include a private payoff for agent i. In this work, we leave

the precise structure and dynamics of s_i^t open.

We assume that each agent i can choose actions a_i^t based on the entire interaction history $H_i^t = (s_i^0, a^0, s_i^1, a^1, ..., s_i^t)$, where $a^\tau = (a_1^\tau, ..., a_m^\tau)$ is the tuple of actions taken by the agents at time τ. Formally, each agent i has a *behaviour* π_i which assigns a probability distribution over actions A_i given a history H_i^t, denoted $\pi_i(H_i^t)$. We use Π_i to denote the infinite and uncountable space of all such behaviours. Note that a behaviour may implement any kind of logic, and it is useful to think of it as a black-box programme.

Given two agents i and j, we use Π_j^i to denote i's *hypothesis space* for j's behaviours. The difference between Π_j^i and Π_j is that $\pi_j^* \in \Pi_j^i$ are defined over H_i^t while $\pi_j \in \Pi_j$ are defined over H_j^t. Since we allow for asymmetric information, any information that is contained in s_j^t but not in s_i^t, denoted s_{j-i}^t, becomes part of the hypothesis space Π_j^i. For example, if s_{j-i}^t contains a private payoff for j, i can hypothesise a payoff as part of its hypothesis for j's behaviour.

Defining a behavioural hypothesis $\pi_j^* \in \Pi_j^i$ as a function $\pi_j^*(H_i^t)$ has two implicit assumptions: firstly, it assumes knowledge of A_j, and secondly, it assumes that the information in s_{j-i}^t is a (deterministic) function of H_i^t. If, on the other hand, we allowed s_{j-i}^t to be stochastic (i.e. a random variable over the interaction history), we would in addition have to hypothesise the random outcome of s_{j-i}^t. In other words, $\pi_j^*(H_i^t)$ would itself be a random variable, which is outside the scope of this work.

4 A METHOD FOR BEHAVIOURAL HYPOTHESIS TESTING

Let i denote our agent and let j denote another agent. Moreover, let $\pi_j^* \in \Pi_j^i$ denote our hypothesis for j's behaviour and let $\pi_j \in \Pi_j$ denote j's true behaviour. The central question we ask is if $\pi_j^* = \pi_j$?

Unfortunately, since we do not know π_j, we cannot directly answer this question. However, at each time t, we know j's past actions $\mathbf{a}_j^t = (a_j^0, ..., a_j^{t-1})$ which were generated by π_j. If we use π_j^* to generate a vector $\hat{\mathbf{a}}_j^t = (\hat{a}_j^0, ..., \hat{a}_j^{t-1})$, where $\hat{a}_j^\tau$ is sampled using $\pi_j^*(H_i^\tau)$, we can formulate the related two-sample problem of whether $\mathbf{a}_j^t$ and $\hat{\mathbf{a}}_j^t$ were generated from the same behaviour, namely π_j^*.

In this section, we propose a general and efficient algorithm to decide this problem. At its core, the algorithm computes a frequentist p-value

$$p = P\left(|T(\tilde{\mathbf{a}}_j^t, \hat{\mathbf{a}}_j^t)| \geq |T(\mathbf{a}_j^t, \hat{\mathbf{a}}_j^t)|\right) \tag{1}$$

where $\tilde{\mathbf{a}}_j^t \sim \delta^t(\pi_j^*) = (\pi_j^*(H_i^0), ..., \pi_j^*(H_i^{t-1}))$. The value of p corresponds to the probability with which we expect to observe a test statistic at least as extreme as $T(\mathbf{a}_j^t, \hat{\mathbf{a}}_j^t)$, under the null-hypothesis $\pi_j^* = \pi_j$. Thus, we reject π_j^* if p is below some "significance level" α.

Algorithm 1

Input: history H_i^t (including observed action a_j^{t-1})
Output: p-value (reject π_j^* if p below some threshold α)
Parameters: hypothesis π_j^*; score functions $z_1, ..., z_K$; $N > 0$
// Expand action vectors
Set $\mathbf{a}_j^t \leftarrow \langle \mathbf{a}_j^{t-1}, a_j^{t-1} \rangle$
Sample $\hat{a}_j^{t-1} \sim \pi_j^*(H_i^{t-1})$; set $\hat{\mathbf{a}}_j^t \leftarrow \langle \hat{\mathbf{a}}_j^{t-1}, \hat{a}_j^{t-1} \rangle$
for $n = 1, ..., N$ **do**
    Sample $\tilde{a}_j^{t-1} \sim \pi_j^*(H_i^{t-1})$; set $\tilde{\mathbf{a}}_j^{t,n} \leftarrow \langle \tilde{\mathbf{a}}_j^{t-1,n}, \tilde{a}_j^{t-1} \rangle$
// Fit skew-normal distribution f
if update parameters? **then**
    Compute $D \leftarrow \left\{ T(\tilde{\mathbf{a}}_j^{t,n}, \hat{\mathbf{a}}_j^t) \mid n = 1, ..., N \right\}$
    Fit ξ, ω, β to D, e.g. using (12)
    Find mode μ from ξ, ω, β
// Compute p-value
Compute $q \leftarrow T(\mathbf{a}_j^t, \hat{\mathbf{a}}_j^t)$ using (2)/(5)
return $p \leftarrow f(q \mid \xi, \omega, \beta) \;/\; f(\mu \mid \xi, \omega, \beta)$

In the following subsections, we describe the test statistic T and its asymptotic properties, and how our algorithm learns the distribution of $T(\tilde{\mathbf{a}}_j^t, \hat{\mathbf{a}}_j^t)$. A summary of the algorithm is given in Algorithm 1.

4.1 TEST STATISTIC

We follow the general approach outlined in Section 1 by which we compute a *score* from a vector of actions and their hypothesised distributions. Formally, we define a *score function* as $z : (A_j)^t \times \Delta(A_j)^t \to \mathbb{R}$, where $\Delta(A_j)$ is the set of all probability distributions over A_j. Thus, $z(\mathbf{a}_j^t, \delta^t(\pi_j^*))$ is the score for observed actions $\mathbf{a}_j^t$ and hypothesised distributions $\delta^t(\pi_j^*)$, and we sometimes abbreviate this to $z(\mathbf{a}_j^t, \pi_j^*)$. We use Z to denote the space of all score functions.

Given a score function z, we define the test statistic T as

$$T(\tilde{\mathbf{a}}_j^t, \hat{\mathbf{a}}_j^t) \;=\; \frac{1}{t} \sum_{\tau=1}^{t} T_\tau(\tilde{\mathbf{a}}_j^\tau, \hat{\mathbf{a}}_j^\tau) \tag{2}$$

$$T_\tau(\tilde{\mathbf{a}}_j^\tau, \hat{\mathbf{a}}_j^\tau) \;=\; z(\tilde{\mathbf{a}}_j^\tau, \pi_j^*) - z(\hat{\mathbf{a}}_j^\tau, \pi_j^*) \tag{3}$$

where $\tilde{\mathbf{a}}_j^\tau$ and $\hat{\mathbf{a}}_j^\tau$ are the τ-prefixes of $\tilde{\mathbf{a}}_j^t$ and $\hat{\mathbf{a}}_j^t$, respectively.

In this work, we assume that z is provided by the user. While formally unnecessary (in the sense that our analysis does not require it), we find it a useful design guideline to interpret a score as a kind of likelihood, such that higher scores suggest higher likelihood of π_j^* being correct. Under this interpretation, a minimum requirement for z should be that it is *consistent*, such that, for any $t > 0$ and $\pi_j^* \in \Pi_j^i$,

$$\pi_j^* \in \Pi^z = \arg\max_{\pi_j' \in \Pi_j^i} \mathbb{E}_{\mathbf{a}_j' \sim \delta^t(\pi_j^*)} \left[z(\mathbf{a}_j', \pi_j') \right] \tag{4}$$

where $\mathbb{E}_\eta$ denotes the expectation under η. This ensures

that if the null-hypothesis $\pi_j^* = \pi_j$ is true, then the score $z(\mathbf{a}_j^t, \pi_j^*)$ is maximised on expectation.

Ideally, we would like a score function z which is *perfect* in that it is consistent and $|\Pi^z| = 1$. This means that π_j^* can maximise $z(\mathbf{a}_j^t, \pi_j^*)$ (where $\mathbf{a}_j^t \sim \delta^t(\pi_j)$) *only* if $\pi_j^* = \pi_j$. Unfortunately, it is unclear if such a score function exists for the general case and how it should look. Even if we restrict the behaviours agents may exhibit, it can still be difficult to find a perfect score function. On the other hand, it is a relatively simple task to specify a small set of score functions $z_1, ..., z_K$ which are consistent but imperfect. (Examples are given in Section 5.) Given that these score functions are consistent, we know that the cardinality $|\cap_k \Pi^{z_k}|$ can only monotonically decrease. Therefore, it seems a reasonable approach to combine multiple imperfect score functions in an attempt to approximate a perfect score function.

Of course, we could simply define z as a linear (or otherwise) combination of $z_1, ..., z_K$. However, this approach is at risk of losing information from the individual scores, e.g. due to commutativity and other properties of the combination. Thus, we instead propose to compare the scores individually. Given score functions $z_1, ..., z_K \in Z$ which are all bounded by the same interval $[a, b] \subset \mathbb{R}$, we redefine T_τ to

$$T_\tau(\tilde{\mathbf{a}}_j^\tau, \hat{\mathbf{a}}_j^\tau) = \sum_{k=1}^{K} w_k \left(z_k(\tilde{\mathbf{a}}_j^\tau, \pi_j^*) - z_k(\hat{\mathbf{a}}_j^\tau, \pi_j^*) \right) \quad (5)$$

where $w_k \in \mathbb{R}$ is a weight for score function z_k. In this work, we set $w_k = \frac{1}{K}$. (We also experiment with alternative weighting schemes in Section 5.) However, we believe that w_k may serve as an interface for useful modifications of our algorithm. For example, Yue et al. (2010) compute weights to increase the power of their specific hypothesis tests.

4.2 ASYMPTOTIC PROPERTIES

The vectors $\mathbf{a}_j^t$ and $\hat{\mathbf{a}}_j^t$ are constructed iteratively. That is, at time t, we observe agent j's past action a_j^{t-1}, which was generated from $\pi_j(H_j^{t-1})$, and set $\mathbf{a}_j^t = \langle \mathbf{a}_j^{t-1}, a_j^{t-1} \rangle$. At the same time, we sample an action $\hat{a}_j^{t-1}$ using $\pi_j^*(H_i^{t-1})$ and set $\hat{\mathbf{a}}_j^t = \langle \hat{\mathbf{a}}_j^{t-1}, \hat{a}_j^{t-1} \rangle$. Assuming the null-hypothesis $\pi_j^* = \pi_j$, will $T(\mathbf{a}_j^t, \hat{\mathbf{a}}_j^t)$ converge in the process?

Unfortunately, T might not converge. This may seem surprising at first glance given that $a_j^{t-1}, \hat{a}_j^{t-1}$ have the same distribution $\pi_j(H_j^{t-1}) = \pi_j^*(H_i^{t-1})$, since $\mathbb{E}_{x,y\sim\psi}[x - y] = 0$ for any distribution ψ. However, there is a subtle but important difference: while $a_j^{t-1}, \hat{a}_j^{t-1}$ have the same distribution, $z_k(\mathbf{a}_j^t, \pi_j^*)$ and $z_k(\hat{\mathbf{a}}_j^t, \pi_j^*)$ may have arbitrarily different distributions. This is because these scores may depend on the entire prefix vectors $\mathbf{a}_j^{t-1}$ and $\hat{\mathbf{a}}_j^{t-1}$, respectively, which means that their distributions may be different if $\mathbf{a}_j^{t-1} \neq \hat{\mathbf{a}}_j^{t-1}$. Fortunately, our algorithm does not require T to converge because it learns the distribution of T during the interaction process, as we will discuss in Section 4.3.

Interestingly, while T may not converge, it can be shown that the fluctuation of T is eventually normally distributed, for any set of score functions $z_1, ..., z_K$ with bound $[a, b]$. Formally, let $\mathbb{E}[T_\tau(\mathbf{a}_j^\tau, \hat{\mathbf{a}}_j^\tau)]$ and $\text{Var}[T_\tau(\mathbf{a}_j^\tau, \hat{\mathbf{a}}_j^\tau)]$ denote the finite expectation and variance of $T_\tau(\mathbf{a}_j^\tau, \hat{\mathbf{a}}_j^\tau)$, where it is irrelevant if $\mathbf{a}_j^\tau, \hat{\mathbf{a}}_j^\tau$ are sampled directly from $\delta^\tau(\pi_j^*)$ or generated iteratively as prescribed above. Furthermore, let $\sigma_t^2 = \sum_{\tau=1}^{t} \text{Var}[T_\tau(\mathbf{a}_j^\tau, \hat{\mathbf{a}}_j^\tau)]$ denote the cumulative variance. Then, the standardised stochastic sum

$$\frac{1}{\sigma_t} \sum_{\tau=1}^{t} T_\tau(\mathbf{a}_j^\tau, \hat{\mathbf{a}}_j^\tau) - \mathbb{E}[T_\tau(\mathbf{a}_j^\tau, \hat{\mathbf{a}}_j^\tau)] \quad (6)$$

will converge in distribution to the standard normal distribution as $t \to \infty$. Thus, T is normally distributed as well.

To see this, first recall that the standard central limit theorem requires the random variables T_τ to be independent and identically distributed. In our case, T_τ are independent in that the random outcome of T_τ has no effect on the outcome of $T_{\tau'}$. However, T_τ and $T_{\tau'}$ depend on different action sequences, and may therefore have different distributions. Hence, we have to show an additional property, commonly known as *Lyapunov's condition* (e.g. Fischer, 2010), which states that there exists a positive integer d such that

$$\lim_{t\to\infty} \frac{\hat{\sigma}_t^{2+d}}{\sigma_t^{2+d}} = 0, \text{ with} \quad (7)$$

$$\hat{\sigma}_t^{2+d} = \sum_{\tau=1}^{t} \mathbb{E}\left[\left| T_\tau(\mathbf{a}_j^\tau, \hat{\mathbf{a}}_j^\tau) - \mathbb{E}[T_\tau(\mathbf{a}_j^\tau, \hat{\mathbf{a}}_j^\tau)] \right|^{2+d} \right]. \quad (8)$$

Since z_k are bounded, we know that T_τ are bounded. Hence, the summands in (8) are uniformly bounded, say by U for brevity. Setting $d = 1$, we obtain

$$\lim_{t\to\infty} \frac{\hat{\sigma}_t^3}{\sigma_t^3} \leq \frac{U\hat{\sigma}_t^2}{\sigma_t^3} = \frac{U}{\sigma_t} \quad (9)$$

The last part goes to zero if $\sigma_t \to \infty$, and hence Lyapunov's condition holds. If, on the other hand, σ_t converges, then this means that the variance of T_τ is zero from some point onward (or that it has an appropriate convergence to zero). In this case, π_j^* will prescribe deterministic action choices for agent j, and a statistical analysis is no longer necessary.

4.3 LEARNING THE TEST DISTRIBUTION

Given that T is eventually normal, it may seem reasonable to compute (1) using a normal distribution whose parameters are fitted during the interaction. However, this fails to recognise that the distribution of T is shaped *gradually* over an extended time period, and that the fluctuation around T can be heavily skewed in either direction until convergence to a normal distribution emerges. Thus, a normal distribution may be a poor fit during this shaping period.

What is needed is a distribution which can represent any normal distribution, and which is flexible enough to faithfully

represent the gradual shaping. One distribution which has these properties is the *skew-normal distribution* (Azzalini, 1985; O'Hagan and Leonard, 1976). Given the PDF ϕ and CDF Φ of the standard normal distribution, the skew-normal PDF is defined as

$$f(x \mid \xi, \omega, \beta) = \frac{2}{\omega}\,\phi\left(\frac{x-\xi}{\omega}\right)\Phi\left(\beta\left(\frac{x-\xi}{\omega}\right)\right) \quad (10)$$

where $\xi \in \mathbb{R}$ is the location parameter, $\omega \in \mathbb{R}^+$ is the scale parameter, and $\beta \in \mathbb{R}$ is the shape parameter. Note that this reduces to the normal PDF for $\beta = 0$, in which case ξ and ω correspond to the mean and standard deviation, respectively. Hence, the normal distribution is a sub-class of the skew-normal distribution.

Our algorithm learns the shifting parameters of f during the interaction process, using a simple but effective sampling procedure. Essentially, we use π_j^* to iteratively generate N additional action vectors $\tilde{\mathbf{a}}_j^{t,1}, ..., \tilde{\mathbf{a}}_j^{t,N}$ in the exact same way as $\hat{\mathbf{a}}_j^t$. The vectors $\tilde{\mathbf{a}}_j^{t,n}$ are then mapped into data points

$$D = \left\{T(\tilde{\mathbf{a}}_j^{t,n}, \hat{\mathbf{a}}_j^t) \mid n = 1, ..., N\right\} \quad (11)$$

which are used to estimate the parameters ξ, ω, β by minimising the negative log-likelihood

$$N\log(\omega) - \sum_{x \in D} \log\phi\left(\frac{x-\xi}{\omega}\right) + \log\Phi\left(\beta\left(\frac{x-\xi}{\omega}\right)\right) \quad (12)$$

whilst ensuring that ω is positive. An alternative is the method-of-moments estimator, which can also be used to obtain initial values for (12). Note that it is usually unnecessary to estimate the parameters at every point in time. Rather, it seems reasonable to update the parameters less frequently as the amount of evidence (i.e. observed actions) grows.

Given the asymmetry of the skew-normal distribution, the semantics of "as extreme as" in (1) may no longer be obvious (e.g. is this with respect to the mean or mode?). In addition, the usual tail-area calculation of the p-value requires the CDF, but there is no closed form for the skew-normal CDF and approximating it is rather cumbersome. To circumvent these issues, we approximate the p-value as

$$p \;\approx\; \frac{f(T(\mathbf{a}_j^t, \hat{\mathbf{a}}_j^t) \mid \xi, \omega, \beta)}{f(\mu \mid \xi, \omega, \beta)} \quad (13)$$

where μ is the mode of the fitted skew-normal distribution. This avoids the asymmetry issue and is easier to compute.

5 EXPERIMENTS

We conducted a comprehensive set of experiments to investigate the accuracy (correct and incorrect rejection), scalability (with number of actions), and sampling complexity of our algorithm. The following three score functions and their combinations were used:

$$z_1(\mathbf{a}_j^t, \pi_j^*) = \frac{1}{t}\sum_{\tau=0}^{t-1} \frac{\pi_j^*(H_i^\tau)[a_j^\tau]}{\max_{a_j \in A_j} \pi_j^*(H_i^\tau)[a_j]}$$

$$z_2(\mathbf{a}_j^t, \pi_j^*) = \frac{1}{t}\sum_{\tau=0}^{t-1} 1 - \mathbb{E}^{a_j \sim}_{\pi_j^*(H_i^\tau)}\left|\pi_j^*(H_i^\tau)[a_j^\tau] - \pi_j^*(H_i^\tau)[a_j]\right|$$

$$z_3(\mathbf{a}_j^t, \pi_j^*) = \sum_{a_j \in A_j} \min\left[\frac{1}{t}\sum_{\tau=0}^{t-1}[a_j^\tau = a_j]_1, \frac{1}{t}\sum_{\tau=0}^{t-1}\pi_j^*(H_i^\tau)[a_j]\right]$$

where $[b]_1 = 1$ if b is true and 0 otherwise. Note that z_1, z_3 are generally consistent (cf. Section 4.1), while z_2 is consistent for $|A_j| = 2$ but not necessarily for $|A_j| > 2$. Furthermore, z_1, z_2, z_3 are all imperfect. The score function z_3 is based on the empirical frequency distribution (cf. Section 1).

The parameters of the test distribution (cf. Section 4.3) were estimated less frequently as t increased. The first estimation was performed at time $t = 1$ (i.e. after observing one action). After estimating the parameters at time t, we waited $\lfloor\sqrt{t}\rfloor - 1$ time steps until the parameters were re-fitted. Throughout our experiments, we used a significance level of $\alpha = 0.01$ (i.e. reject π_j^* if the p-value is below 0.01).

5.1 RANDOM BEHAVIOURS

In the first set of experiments, the behaviour spaces Π_i, Π_j and hypothesis space Π_j^i were restricted to "random" behaviours. Each random behaviour is defined by a sequence of random probability distributions over A_j. The distributions are created by drawing uniform random numbers from $(0, 1)$ for each action $a_j \in A_j$, and subsequent normalisation so that the values sum up to 1.

Random behaviours are a good baseline for our experiments because they are usually hard to distinguish. This is due to the fact that the entire set A_j is always in the support of the behaviours, and since they do not react to any past actions. These properties mean that there is little structure in the interaction that can be used to distinguish behaviours.

We simulated 1000 interaction processes, each lasting 10000 time steps. In each process, we randomly sampled behaviours $\pi_i \in \Pi_i$, $\pi_j \in \Pi_j$ to control agents i and j, respectively. In half of these processes, we used a correct hypothesis $\pi_j^* = \pi_j$. In the other half, we sampled a random hypothesis $\pi_j^* \in \Pi_j^i$ with $\pi_j^* \neq \pi_j$. We repeated each set of simulations for $|A_j| = 2, 10, 20$ (with $|A_i| = |A_j|$) and $N = 10, 50, 100$ (cf. Section 4.3).

5.1.1 Accuracy & Scalability

Figure 1 shows the average accuracy of our algorithm (for $N = 50$), by which we mean the average percentage of time steps in which the algorithm made correct decisions (i.e. no reject if $\pi_j^* = \pi_j$; reject if $\pi_j^* \neq \pi_j$). The x-axis shows

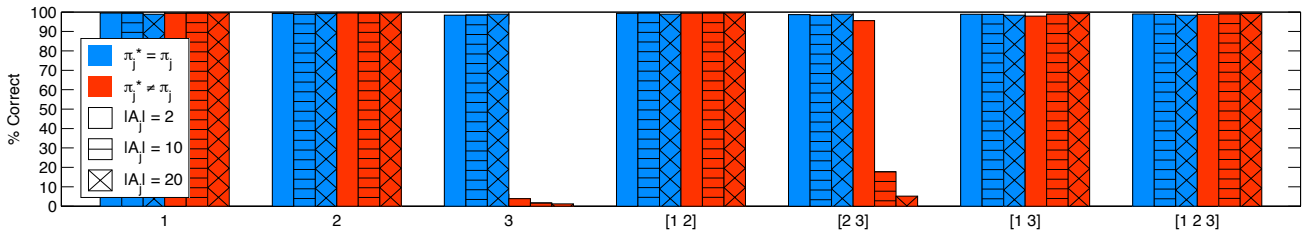
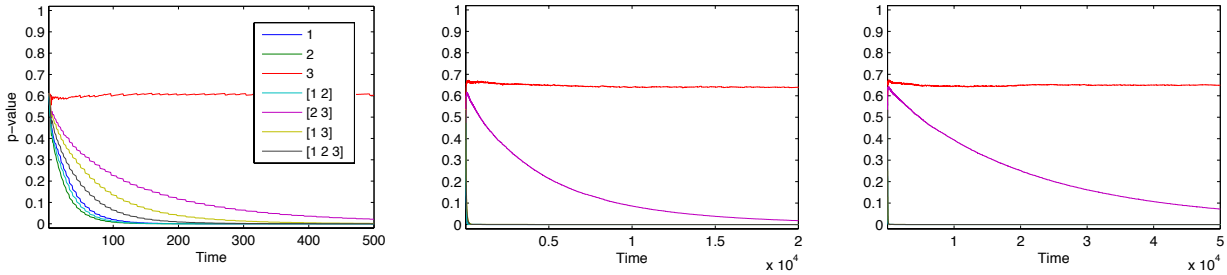


Figure 1: Average accuracy with random behaviours, for $N = 50$ and $|A_j| = 2, 10, 20$. Results averaged over 500 processes with 10000 time steps, for $\pi_j^* = \pi_j$ and $\pi_j^* \neq \pi_j$ each. X-axis shows score functions z_k used in test statistic.



(a) $|A_j| = 2$ (b) $|A_j| = 10$ (c) $|A_j| = 20$

Figure 2: Average p-values with random behaviours, for $N = 50$ and $\pi_j^* \neq \pi_j$ (i.e. hypothesis wrong). Results averaged over 500 processes. Legend shows score functions z_k used in test statistic.

the combination of score functions used to compute the test statistic (e.g. [1 2] means that we combined z_1, z_2).

The results show that our algorithm achieved excellent accuracy, often bordering the 100% mark. They also show that the algorithm scaled well with the number of actions, with no degradation in accuracy. However, there were two exceptions to these observations: Firstly, using z_3 resulted in very poor accuracy for $\pi_j^* \neq \pi_j$. Secondly, the combination of z_2, z_3 scaled badly for $\pi_j^* \neq \pi_j$.

The reason for both of these exceptions is that z_3 is not a good scoring scheme for random behaviours. The function z_3 quantifies a similarity between the empirical frequency distribution and the averaged hypothesised distributions. For random behaviours (as defined in this work), both of these distributions will converge to the uniform distribution. Thus, under z_3, any two random behaviours will eventually be the same, which explains the low accuracy for $\pi_j^* \neq \pi_j$.

As can be seen in Figure 1, the inadequacy of z_3 is solved when adding any of the other score functions z_1, z_2. These functions add discriminative information to the test statistic, which technically means that the cardinality $|\Pi^z|$ in (4) is reduced. However, in the case of $[z_2, z_3]$, the converge is substantially slower for higher $|A_j|$, meaning that more evidence is needed until π_j^* can be rejected. Figure 2 shows how a higher number of actions affects the average convergence rate of p-values computed with z_2, z_3.

In addition to the score functions z_k, a central aspect for the convergence of p-values are the corresponding weights w_k (cf. (5)). As mentioned in Section 4.1, we use uniform weights $w_k = \frac{1}{K}$. However, to show that the weighting is no trivial matter, we repeated our experiments with four alternative weighting schemes: Let $z_k^\tau = z_k(\tilde{\mathbf{a}}_j^\tau, \pi_j^*) - z_k(\hat{\mathbf{a}}_j^\tau, \pi_j^*)$ denote the summands in (5). The weighting schemes `truemax` / `truemin` assign $w_k = 1$ for the first k that maximises / minimises $|z_k^\tau|$, and 0 otherwise. Similarly, the weighting schemes `max` / `min` assign $w_k = 1$ for the first k that maximises / minimises z_k^τ, and 0 otherwise.

Figures 3 and 4 show the results for `truemax` and `truemin`. As can be seen in the figures, `truemax` is very similar to uniform weights while `truemin` improves the convergence for $[z_2, z_3]$ but compromises elsewhere. The results for `max` and `min` are very similar to those of `truemin` and `truemax`, respectively, hence we omit them.

Finally, we recomputed all accuracies using a more lenient significance level of $\alpha = 0.05$. As could be expected, this marginally decreased and increased (i.e. by a few percentage points) the accuracy for $\pi_j^* = \pi_j$ and $\pi_j^* \neq \pi_j$, respectively. Overall, however, the results were very similar to those obtained with $\alpha = 0.01$.

5.1.2 Sampling Complexity

Recall that N specifies the number of sampled action vectors $\tilde{\mathbf{a}}_j^{t,n}$ used to learn the distribution of the test statistic (cf. Section 4.3). In the previous section, we reported results for $N = 50$. In this section, we investigate differences in accuracy for $N = 10, 50, 100$.

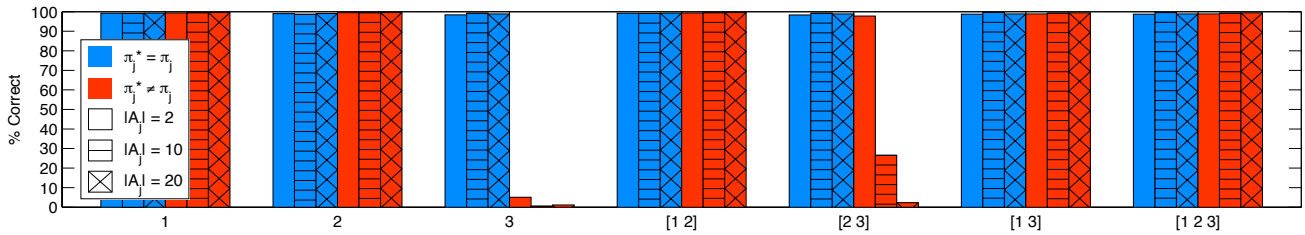


Figure 3: Average accuracy with random behaviours, for $N = 50$ and $|A_j| = 2, 10, 20$. X-axis shows score functions z_k used in test statistic. Weights w_k computed using `truemax` weighting.

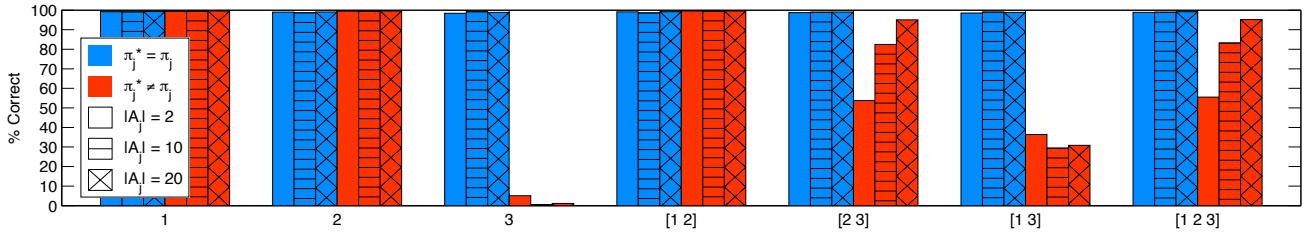


Figure 4: Average accuracy with random behaviours, for $N = 50$ and $|A_j| = 2, 10, 20$. X-axis shows score functions z_k used in test statistic. Weights w_k computed using `truemin` weighting.

Figures 5 and 6 show the differences for $|A_j| = 2, 20$, respectively. (The figure for $|A_j| = 10$ was virtually the same as the one for $|A_j| = 20$, except with minor improvements in accuracy for the $[z_2, z_3]$ cluster. Hence, we omit it here.) As can be seen, there were improvements of up to 10% from $N = 10$ to $N = 50$, and no (or very marginal) improvements from $N = 50$ to $N = 100$. This was observed for all $|A_j| = 2, 10, 20$, and all constellations of score functions. The fact that $N = 50$ was sufficient even for $|A_j| = 20$ is remarkable, since, under random behaviours, there are 20^t possible action vectors to sample at any time t.

We also compared the learned skew-normal distributions and found that they fitted the data very well. Figures 7 and 8 show the histograms and fitted skew-normal distributions for two example processes after 1000 time steps. In Figure 8, we deliberately chose an example in which the learned distribution was maximally skewed for $N = 10$, which is a sign that N was too small. Nonetheless, in the majority of the processes, the learned distribution was only moderately skewed and our algorithm achieved an average accuracy of 90% even for $N = 10$. Moreover, if one wants to avoid maximally skewed distributions, one can simply restrict the parameter space when fitting the skew-normal (specifically, the shape parameter β; cf. Section 4.3).

The flexibility of the skew-normal distribution was particularly useful in the early stages of the interaction, in which the test statistic typically does not follow a normal distribution. Figure 9 shows the test distribution for an example process after 10 time steps, using z_2 for the test statistic and $N = 100$ (the histogram was created using $N = 10000$). The learned skew-normal approximated the true test distribution very closely. Note that, in such examples, the normal and Student distributions do not produce good fits.

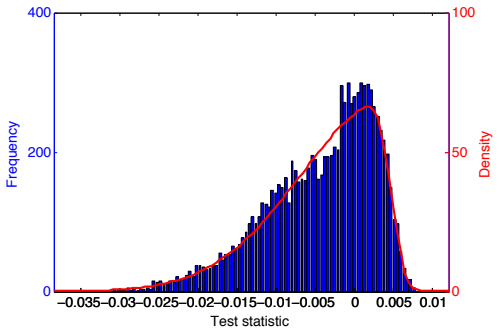


Figure 9: True test distribution for z_2 (histogram) and learned skew-normal distribution (red curve) after 10 time steps, with $|A_j| = 10$ and $N = 100$.

Our implementation of the algorithm performed all calculations as iterative updates (except for the skew-normal fitting). Hence, it used little (fixed) memory and had very low computation times. For example, using all three score functions and $|A_j| = 20$, $N = 100$, one cycle in the algorithm (cf. Algorithm 1) took on average less than 1 millisecond without fitting the skew-normal parameters, and less than 10 milliseconds when fitting the skew-normal parameters (using an off-the-shelf Simplex-optimiser with default parameters). The times were measured using Matlab R2014a on a Unix machine with a 2.6 GHz Intel Core i5 processor.

5.2 ADAPTIVE BEHAVIOURS

We complemented the "structure-free" interaction of random behaviours by conducting analogous experiments with

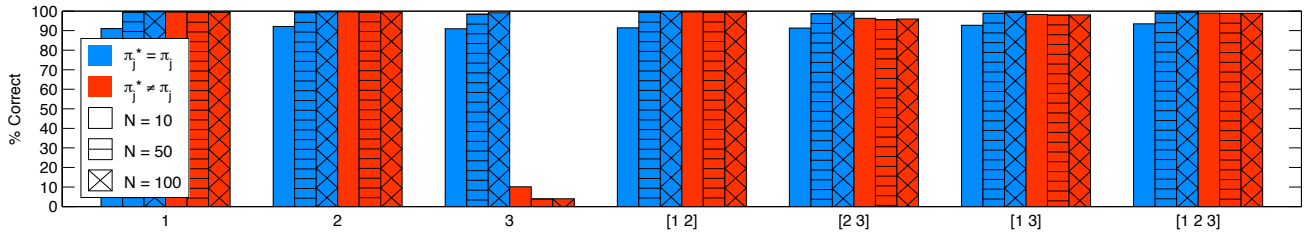


Figure 5: Average accuracy with random behaviours, for $|A_j| = 2$ and $N = 10, 50, 100$. Results averaged over 500 processes with 10000 time steps, for $\pi_j^* = \pi_j$ and $\pi_j^* \neq \pi_j$ each. X-axis shows score functions z_k used in test statistic.

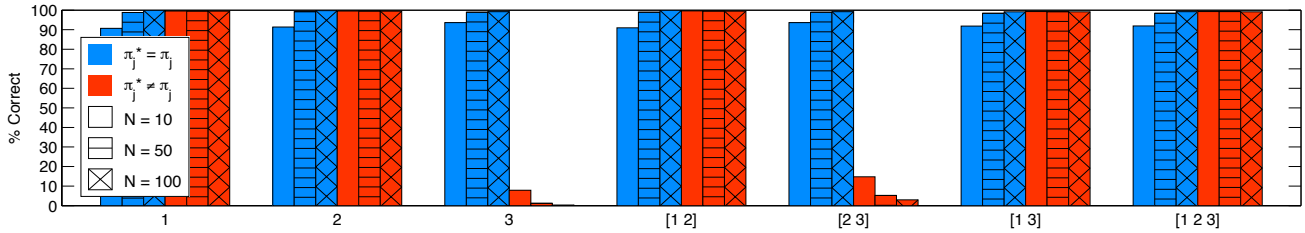


Figure 6: Average accuracy with random behaviours, for $|A_j| = 20$ and $N = 10, 50, 100$. Results averaged over 500 processes with 10000 time steps, for $\pi_j^* = \pi_j$ and $\pi_j^* \neq \pi_j$ each. X-axis shows score functions z_k used in test statistic.

three additional classes of behaviours. Specifically, we used a benchmark framework specified by Albrecht et al. (2015) which consists of 78 distinct 2×2 matrix games and three methods to automatically generate sets of behaviours for any given game. The three behaviour classes are Leader-Follower-Trigger Agents (LFT), Co-Evolved Decision Trees (CDT), and Co-Evolved Neural Networks (CNN). These classes cover a broad spectrum of possible behaviours, including fully deterministic (CDT), fully stochastic (CNN), and hybrid (LFT) behaviours. Furthermore, all generated behaviours are *adaptive* to varying degrees (i.e. they adapt their action choices based on the other player's choices). We refer to Albrecht et al. (2015) for a more detailed description of these classes (we used the same parameter settings).

The following experiments were performed for each behaviour class, using identical randomisation: For each of the 78 games, we simulated 10 interaction processes, each lasting 10000 time steps. For each process, we randomly sampled behaviours $\pi_i \in \Pi_i, \pi_j \in \Pi_j$ to control agents i and j, respectively, where Π_i, Π_j (and Π_j^i) were restricted to the same behaviour class. In half of these processes, we used a correct hypothesis $\pi_j^* = \pi_j$, and in the other half, we sampled a random hypothesis $\pi_j^* \in \Pi_j^i$ with $\pi_j^* \neq \pi_j$. As before, we repeated each simulation for $N = 10, 50, 100$ and all constellations of score functions, but found that there were virtually no differences. Hence, in the following, we report results for $N = 50$ and the $[z_1, z_2, z_3]$ cluster.

Figure 10 shows the average accuracy achieved by our algorithm for all three behaviour classes. While the accuracy for $\pi_j^* = \pi_j$ was generally good, the accuracy for $\pi_j^* \neq \pi_j$ was mixed. Note that this was not merely due to the fact that the score functions were imperfect (cf. Section 4.1), since we

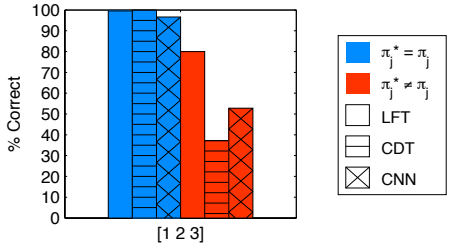


Figure 10: Average accuracy for behaviour classes LFT, CDT, CNN ($N = 50$). Π_i and Π_j restricted to same class.

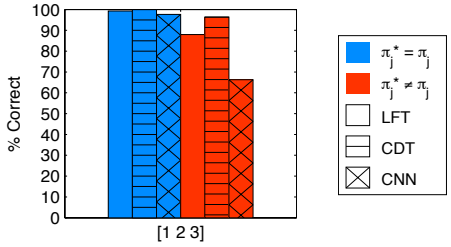
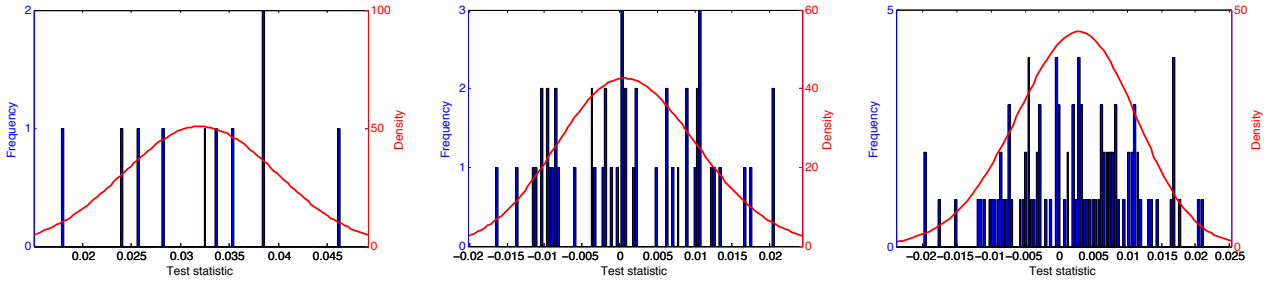


Figure 11: Average accuracy for behaviour classes LFT, CDT, CNN ($N = 50$). Π_i set to random behaviours.

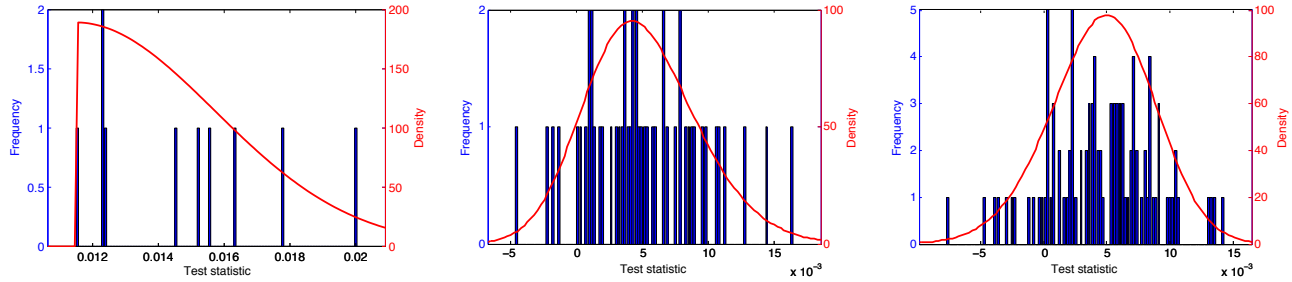
obtained the same results for all combinations. Rather, this reveals an inherent limitation of our approach, which is that *we do not actively probe aspects of the hypothesis* π_j^*. In other words, our algorithm performs statistical hypothesis tests based only on evidence that was generated by π_i.

To illustrate this, it is useful to consider the tree structure of behaviours in the CDT class. Each node in a tree π_j corresponds to a past action taken by π_i. Depending on how π_i chooses actions, we may only ever see a subset of the



(a) $N = 10$ (b) $N = 50$ (c) $N = 100$

Figure 7: Example histograms and fitted skew-normal distributions (red curve) after 1000 time steps, for random behaviours with $|A_j| = 10$ and $N = 10, 50, 100$. Using score function z_1 in test statistic.



(a) $N = 10$ (b) $N = 50$ (c) $N = 100$

Figure 8: Example histograms and fitted skew-normal distributions (red curve) after 1000 time steps, for random behaviours with $|A_j| = 10$ and $N = 10, 50, 100$. Using score functions z_1, z_2, z_3 in test statistic.

entire tree that defines π_j. However, if our hypothesis π_j^* differs from π_j only in the unseen aspects of π_j, then there is no way for our algorithm to differentiate the two. Hence the asymmetry in accuracy for $\pi_j^* = \pi_j$ and $\pi_j^* \neq \pi_j$. Note that this problem did not occur in random behaviours because, there, all aspects are eventually visible.

Following this observation, we repeated the same experiments but restricted Π_i to random behaviours, with the goal of exploring π_j^* more thoroughly. As shown in Figure 11, this led to significant improvements in accuracy, especially for the CDT class. Nonetheless, choosing actions purely randomly may not be a sufficient probing strategy, hence the accuracy for CNN was still relatively low. For CNN, this was further complicated by the fact that two neural networks π_j, π_j' may formally be different ($\pi_j \neq \pi_j'$) but have essentially the same action probabilities (with extremely small differences). Hence, in such cases, we would require much more evidence to distinguish the behaviours.

6 CONCLUSION

We hold the view that if an intelligent agent is to interact effectively with other agents whose behaviours are unknown, it will have to hypothesise what these agents might be doing *and* contemplate the truth of its hypotheses, such that appropriate measures can be taken if they are deemed false. In this spirit, we presented a novel algorithm which decides this question in the form of a frequentist hypothesis test. The algorithm can incorporate multiple statistical criteria into the test statistic and learns the test distribution during the interaction process, with asymptotic correctness guarantees. We presented results from a comprehensive set of experiments, showing that our algorithm achieved high accuracy and scalability at low computational costs.

There are several directions for future work: To bring some structure into the space of score functions, we introduced the concepts of consistency and perfection as minimal and ideal properties. However, more research is needed to understand precisely what properties a useful score function should satisfy, and whether the concept of perfection is feasible or even necessary in the general case. Furthermore, we used uniform weights to combine the computed scores into a test statistic, and we also experimented with alternative weighting schemes to show that the weighting can have a substantial effect on convergence rates. However, further research is required to understand the effect of weights on decision quality and convergence.

Finally, in this work, we assumed that the behaviour of the other agent (j) could be described as a function of the information available to our agent (i). An important extension would be to also account for information that cannot be deterministically derived from our observations, especially in the context of robotics where observations are often described as random variables.

References

S.V. Albrecht and S. Ramamoorthy. On convergence and optimality of best-response learning with policy types in multiagent systems. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence*, pages 12–21, 2014.

S.V. Albrecht, J.W. Crandall, and S. Ramamoorthy. An empirical study on the practical impact of prior beliefs over policy types. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 1988–1994, 2015.

A. Azzalini. A class of distributions which includes the normal ones. *Scandinavian Journal of Statistics*, 12:171–178, 1985.

I.V. Basawa and D.J. Scott. Efficient tests for stochastic processes. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 21–31, 1977.

M.J. Bayarri and J.O. Berger. P values for composite null models. *Journal of the American Statistical Association*, 95(452):1127–1142, 2000.

J.O. Berger and T. Sellke. Testing a point null hypothesis: the irreconcilability of p values and evidence (with discussion). *Journal of the American Statistical Association*, 82:112–122, 1987.

G.E.P. Box. Sampling and Bayes' inference in scientific modelling and robustness. *Journal of the Royal Statistical Society. Series A (General)*, pages 383–430, 1980.

G.W. Brown. Iterative solution of games by fictitious play. *Activity Analysis of Production and Allocation*, 13(1):374–376, 1951.

S. Carberry. Techniques for plan recognition. *User Modeling and User-Adapted Interaction*, 11(1-2):31–48, 2001.

D. Carmel and S. Markovitch. Exploration strategies for model-based learning in multi-agent systems: Exploration strategies. *Autonomous Agents and Multi-Agent Systems*, 2(2):141–172, 1999.

E. Charniak and R.P. Goldman. A Bayesian model of plan recognition. *Artificial Intelligence*, 64(1):53–79, 1993.

E.M. Clarke, O. Grumberg, and D.A. Peled. *Model Checking*. MIT Press, 1999.

V. Conitzer and T. Sandholm. AWESOME: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. *Machine Learning*, 67(1-2):23–43, 2007.

D.R. Cox. The role of significance tests (with discussion). *Scandinavian Journal of Statistics*, 4:49–70, 1977.

H. Fischer. *A History of the Central Limit Theorem: From Classical to Modern Probability Theory*. Springer Science & Business Media, 2010.

R.A. Fisher. *The Design of Experiments*. Oliver & Boyd, 1935.

D.P. Foster and H.P. Young. Learning, hypothesis testing, and Nash equilibrium. *Games and Economic Behavior*, 45(1):73–96, 2003.

A. Gelman and C.R. Shalizi. Philosophy and the practice of Bayesian statistics. *British Journal of Mathematical and Statistical Psychology*, 66(1):8–38, 2013.

I. Gilboa and D. Schmeidler. *A Theory of Case-Based Decisions*. Cambridge University Press, 2001.

P.J. Gmytrasiewicz and P. Doshi. A framework for sequential planning in multiagent settings. *Journal of Artificial Intelligence Research*, 24(1):49–79, 2005.

K.G. Larsen and A. Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94(1):1–28, 1991.

X.-L. Meng. Posterior predictive p-values. *The Annals of Statistics*, pages 1142–1160, 1994.

A. O'Hagan and T. Leonard. Bayes estimation subject to uncertainty about parameter constraints. *Biometrika*, 63 (1):201–203, 1976.

D.B. Rubin. Bayesianly justifiable and relevant frequency calculations for the applied statistician. *The Annals of Statistics*, 12(4):1151–1172, 1984.

D. Ryabko and B. Ryabko. On hypotheses testing for ergodic processes. In *Proceedings of IEEE Information Theory Workshop*, pages 281–283, 2008.

A. Vehtari and J. Ojanen. A survey of Bayesian predictive methods for model assessment, selection and comparison. *Statistics Surveys*, 6:142–228, 2012.

Y. Yue, Y. Gao, O. Chapelle, Y. Zhang, and T. Joachims. Learning more powerful test statistics for click-based retrieval evaluation. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 507–514, 2010.