Web Intelligence and Agent Systems: An International Journal 0 (2009) 1–0
IOS Press

Model Identification in Interactive Influence Diagrams Using Mutual Information

Yifeng Zeng [a,*] and Prashant Doshi [b]
[a] *Dept. of Computer Science*
Aalborg University, Denmark
DK-9220 Aalborg, Denmark
E-mail: yfzeng@cs.aau.dk
[b] *Dept. of Computer Science*
University of Georgia, U.S.A.
Athens, GA 30602
E-mail: pdoshi@cs.uga.edu

Abstract. Interactive influence diagrams (I-IDs) offer a transparent and intuitive representation for the decision-making problem in multiagent settings. They ascribe procedural models such as influence diagrams and I-IDs to model the behavior of other agents. Procedural models offer the benefit of understanding how others arrive at their behaviors. Accurate behavioral models of others facilitate optimal decision-making in multiagent settings. However, identifying the true models of other agents is a challenging task. Given the assumption that the true model of the other agent lies within the set of models that we consider, we may utilize standard Bayesian learning to update the likelihood of each model given the observation histories of others' actions. However, as model spaces are often bounded, the true models of others may not be present in the model space. We then seek to identify models that are *relevant* to the observed behaviors of others and show how the agent may learn to identify these models. We evaluate the performance of our method on three repeated games and provide theoretical and empirical results in support.

Keywords: Influence Diagrams, Mutual Information, Bayesian Learning, Opponent Modeling

1. Introduction

Interactive influence diagrams (I-IDs) [7] are graphical models for decision making in uncertain multiagent settings. I-IDs generalize influence diagrams (IDs) [21] to make them applicable to settings shared with other agents, who may themselves act, observe and update their beliefs. I-IDs and their sequential counterparts, interactive dynamic influence diagrams (I-DIDs) [7], contribute to a growing line of work that includes multiagent influence diagrams (MAID) [12], and more recently, networks of influence diagrams (NID) [9]. All of these formalisms seek to explicitly and transparently model the structure that is often present in real-world problems by decomposing the situation into chance and decision variables, and the dependencies between the variables.

I-IDs ascribe *procedural* models to other agents – these may be IDs, Bayesian networks (BN), or I-IDs themselves leading to recursive modeling. Besides providing intuitive reasons for the strategies, procedural knowledge may help preclude certain strategies of others, deeming them impossible because of the structure of the environment. As agents act and make observations, beliefs over others' models are updated. With the implicit assumption that the true model of other is contained in the model space, I-IDs use Bayesian learning to update beliefs, which gradually converge.

However, in the absence of this assumption, Bayesian learning is not guaranteed to converge and in fact, may become undefined. This is significant because though there are uncountably infinite numbers of agent func-

[*]Corresponding author. E-mail: yfzeng@cs.aau.dk

tions, there are only countable computable models. Hence, theoretically it is likely that an agent's true model may not be within the computable model space. This insight is not new – it motivated Suryadi and Gmytrasiewicz [20] to modify the IDs ascribed to others when observations of other's behaviors were inconsistent with the model space during model identification.

An alternative to considering candidate models is to restrict the models to those represented using a modeling language and directly learn, possibly approximate, models expressed in the language. For example, Carmel and Markovitch [3] learn finite state automatons to model agents' strategies, and Saha *et al.* [17] learn Chebychev polynomials to approximate agents' decision functions. However, the representations are non-procedural and the learning problems complex.

In this article, we consider the realistic case that the true model may not be within the bounded model space of the other agent in an I-ID. In this context, we present a technique that identifies a model or a weighted combination of models whose predictions are *relevant* to the observed action history. Using previous observations of others' actions and predictions by candidate models, we learn how the predictions may relate to the observation history. In other words, we learn to *classify* the predictions of the candidate models using the previous observation history as the training set. Thus, we seek the hidden function that possibly relates the candidate models to the true model.

We then update the likelihoods of the candidate models. As a Bayesian update may be inadequate, we utilize the similarity between the predictions of a candidate model and the observed actions as the likelihood of the model. In particular, we measure the *mutual information* of the predicted actions by a candidate model and the observed actions. This provides a natural measure of the dependence between the candidate and true models, possibly due to some shared behavioral aspects. We theoretically analyze the properties and empirically evaluate the performance of our approach on multiple problem domains modeled using I-IDs. We demonstrate that an agent utilizing the approach gathers larger rewards on average as it better predicts the actions of others.

The remainder of this article is structured as follows: in Section 2, we analyze the related work. In Section 3, we briefly review the graphical model of I-ID that underlies our work, and discuss Bayesian learning in I-IDs. In Section 4, we formally propose an information-theoretic method for model identification and provide relevant theoretical results. We then offer, in Section 5, experimental results that demonstrate the performance of our proposed technique comparing it with other approaches on three repeated games. Section 6 concludes this article with a discussion and remarks on future work.

2. Related Work

The benefits of utilizing graphical models for representing agent interactions have been recognized previously. Suryadi and Gmytrasiewicz [20] used IDs to model other agents and Bayesian learning to update the distributions over the models based on observed behavior. Additionally, they also consider the case where none of the candidate models reflect the observed behavior. In this situation, Suryadi and Gmytrasiewicz show how certain aspects of the IDs may be altered to better reflect the observed behavior. In comparison, we seek to find the underlying dependencies that may exist between candidate models and the true model.

More recently, MAIDs [12] and NIDs [9] extend IDs to multiagent settings. MAIDs objectively analyze the game, efficiently computing the Nash equilibrium profile by exploiting the independence structure. NIDs extend MAIDs to include agents' uncertainty over the game being played and over models of the other agents. MAIDs provide an analysis of the game from an external viewpoint and the applicability of both is limited to single step play in static games. NIDs collapse into MAIDs and both focus on solutions that are in Nash equilibrium. While I-IDs could be seen as NIDs, they model the subjective decision-making problem of an agent, and their dynamic extensions, I-DIDs [7] model interactions that are extended over time.

Bayesian learning is widely used for identifying agents' strategies in multiagent interactions. Gmytrasiewicz et al. [10] used a Bayesian method to update the beliefs about agent models within the recursive modeling framework. Zeng and Sycara [22] learned agents' behaviors through Bayesian updates in a negotiation process. A more sophisticated framework using Bayesian learning was built to learn opponent models in automated negotiation [11]. Both these applications demonstrate the effectiveness of Bayesian learning but rely on a hypothesis that the strategy of an opponent resides in a preference profiler. Recently, Madsen and Jensen [14] implemented opponent modeling using dynamic influence diagrams. Their experimental results on a *Grid* problem illustrate that Bayesian learning becomes undefined when the true model of an opponent does not fall within the predefined model space.

Extensions of the minimax algorithm [1,19] to incorporate different opponent strategies (rather than just being rational) have also been investigated. However, this line of work focuses on improving the applicability of the minimax algorithm and uses agent functions as models. It assumes that the true model of the opponent is within the set of candidate models. In a somewhat different approach, Saha et al. [17] ascribe orthogonal Chebychev polynomials as agent functions. They provide an algorithm to learn the coefficients of the polynomials using the observation history. However, both the degree and the number of polynomials is fixed a'priori thereby bounding the model space, and a best fit function is obtained.

3. Background

We briefly describe interactive influence diagrams (I-IDs) [7] for modeling two-agent interactions and illustrate their application using a simple example. We also discuss Bayesian learning for identifying models in I-IDs and point out a subtle limitation which is of significance.

3.1. Overview of I-IDs

We begin by discussing the syntax of I-IDs and the procedure for solving them.

3.1.1. Syntax and Solution

In addition to the usual chance, decision, and utility nodes, I-IDs include a new type of node called the *model* node (hexagon in Fig. 1(a)). The probability distribution over the model node represents an agent, say i's, belief over the candidate models of the other agent j. In addition to the model node, I-IDs differ from IDs by having a chance node, A_j, that represents the distribution over the other agent j's actions, and a dashed link, called a *policy link*.

The model node $M_{j,l-1}$ contains as its values the alternative computational models ascribed by i to the other agent j at a lower level, $l-1$. Formally, we denote a model of j as $m_{j,l-1}$. A model in the model node, for example, may itself be an I-ID, in which case the recursion terminates when a model is an ID or a BN. If $m_{j,l-1}$ is an I-ID, $m_{j,l-1} = \langle b_{j,l-1}, \hat{\theta}_j \rangle$, where $b_{j,l-1}$ is the belief of agent j and $\hat{\theta}_j$ is the agent's

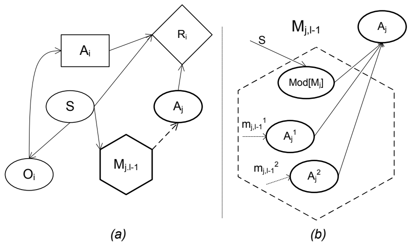


Fig. 1. (a) Generic I-ID for agent i situated with one other agent j. The hexagon is the model node whose structure we show in (b). Members of model node may be IDs, BNs or I-IDs themselves (m_j^1, m_j^2; not shown here for simplicity) whose decision nodes are mapped to the corresponding chance nodes (A_j^1, A_j^2).

frame encompassing the action, observation, and utility nodes. We observe that the model node and the dashed policy link that connects it to the chance node, A_j, could be represented as shown in Fig. 1(b). Once an I-ID or ID of j is solved and the optimal decisions are determined, the decision node is transformed into a chance node[1]. The chance node has the decision alternatives as possible states and a probability distribution over the states. Specifically, if OPT is the set of optimal actions obtained by solving the I-ID (or ID), then $Pr(a_j \in A_j^1) = \frac{1}{|OPT|}$ if $a_j \in OPT$, 0 otherwise. The different chance nodes (A_j^1, A_j^2), one for each model, and additionally, the chance node labeled $Mod[M_j]$ form the parents of the chance node, A_j. Thus, there are as many action nodes as the number of models in the support of agent i's belief. The states of $Mod[M_j]$ denote the different models of j. The distribution over $Mod[M_j]$ is i's belief over j's candidate models (model weights) given the physical state S. The conditional probability distribution (CPD) of the chance node, A_j, is a *multiplexer* that assumes the distribution of each of the action nodes (A_j^1, A_j^2) depending on the state of $Mod[M_j]$. In other words, when $Mod[M_j]$ has the state m_j^1, the chance node A_j assumes the distribution of A_j^1, and A_j assumes the distribution of A_j^2 when $Mod[M_j]$ has the state m_j^2.

Solution of level l I-ID proceeds in a bottom-up manner, and is implemented recursively in Fig.2. We start by solving the lower level models, which are level $l-1$ I-ID or level 0 ID (line 3). Their solutions provide probability distributions over the other agents' actions, which are entered in the corresponding chance nodes found in the model node of the I-ID (line

[1] If j's model is a BN, a chance node representing j's decisions will be directly mapped into a chance node in the model node.

Table 1
PG game with punishment. Based on punishment, P, and marginal return, c_i, agents may choose to contribute than defect.

i, j	FC	PC	D
FC	$2c_iX_T$, $2c_jX_T$	$\frac{3}{2}X_Tc_i - \frac{1}{2}c_p$, $\frac{1}{2}X_T + \frac{3}{2}X_Tc_j - \frac{1}{2}P$	$c_iX_T - c_p$, $X_T + c_jX_T - P$
PC	$\frac{1}{2}X_T + \frac{3}{2}X_Tc_i - \frac{1}{2}P$, $\frac{3}{2}X_Tc_j - \frac{1}{2}c_p$	$\frac{1}{2}X_T + c_iX_T$, $\frac{1}{2}X_T + c_jX_T$	$\frac{1}{2}X_T + \frac{1}{2}c_iX_T - \frac{1}{2}P$, $X_T + \frac{1}{2}c_jX_T - P$
D	$X_T + c_iX_T - P$, $c_jX_T - c_p$	$X_T + \frac{1}{2}c_iX_T - P$, $\frac{1}{2}X_T + \frac{1}{2}c_jX_T - \frac{1}{2}P$	X_T, X_T

4). The mapping from the candidate models' decision nodes to the chance nodes is carried out so that actions with the largest value in the decision node are assigned uniform probabilities in the chance node while the rest are assigned zero probability. Given the distributions over the actions within the different chance nodes (one for each model of the other agent), the I-ID is transformed into a traditional ID. During the transformation, the CPD of the node, A_j, is populated such that the node assumes the distribution of each of the chance nodes depending on the state of the node, $Mod[M_{j,l-1}]$ (line 5). As we mentioned previously, the states of the node $Mod[M_{j,l-1}]$ denote the different models of the other agent, and its distribution is agent i's belief over the models of j conditioned on the physical state. The transformed I-ID is a traditional ID that may be solved using the standard expected utility maximization method (line 6) [18].

I-ID SOLUTION(level $l \geq 1$ I-ID or level 0 ID)

1. **If** $l \geq 1$ **then**
2. **For each** $m^p_{j,l-1}$ **in** $Mod[M_{j,l-1}]$ **do**
3. Recursively call algorithm with the $l-1$ I-ID (or ID) that represents $m^p_{j,l-1}$
4. Map the decision node of the solved I-ID (or ID), $OPT(m^p_{j,l-1})$, to the chance node A^p_j
5. Establish CPD of the chance node A_j in the I-ID
6. Apply the standard expected utility maximization method to solve the transformed I-ID

Fig. 2. Algorithm for solving a level $l \geq 1$ I-ID or level 0 ID

3.1.2. Illustration

We illustrate I-IDs using an example application to the public good (PG) game with punishment (Table 1) explained in detail in [8]. Two agents, i and j, must either contribute some resource to a public pot or keep it for themselves. To make the game more interesting, we allow agents to contribute the full (FC) or a partial (PC) portion of their resources though they could defect (D) and not make any contribution. The value of resources in the public pot is shared by the agents regardless of their actions and is discounted by c_i for each agent i, where $c_i \in (0, 1)$ is the marginal private return. As defection is a dominating action, we introduce a punishment P to penalize the defecting agents and to promote contribution. Additionally, a non-zero cost c_p of punishing is incurred by the contributing agents. For simplicity, we assume each agent has the same amount, X_T, of private resources and a partial contribution is $\frac{1}{2}X_T$.

We let agents i and j play the PG game repeatedly a finite number of times and aim for largest average rewards. After a round of play, agents observe the simultaneous actions of their opponents. Except for the observation of actions, no additional information is shared between the agents. As discovered in field experiments with humans [2], different types of agents play PG differently. To act rationally, i ascribes candidate behavioral models to j. We assume the models are graphical taking the form of IDs and BNs.

For illustration, let agent i consider four models of j (m^1_j, m^2_j, m^3_j, and m^4_j) in the model node at time t, as shown in Fig. 3. The first two models, m^1_j and m^2_j, are simple IDs where the chance node $A_{i,\langle 1,\cdots,t-1\rangle}$ represents the frequencies of the different actions of agent i in the game history (from 1 to time $t-1$). However, the two IDs have different reward functions in the value node. The model m^1_j has a typical low marginal private return, c_j, and represents a reciprocal agent who contributes only when it expects the other agent to contribute. The model m^2_j has a high c_j and represents an altruistic agent who prefers to contribute during the play. The third model, m^3_j, is a BN representing that j's behavior relies on its own action in the previous time step ($A_{j,t-1}$) and i's previous action ($A_{i,t-1}$). m_j^4 represents a more sophisticated decision process. Agent j considers not only its own and i's actions at time $t-1$ (chance nodes $A_{i,t-1}$ and $A_{j,t-1}$), but also agent i's actions at time $t-2$ ($A_{i,t-2}$). It indicates that j relies greatly on the history of the interaction to choose its actions at time t. We point out that these four models reflect typical thinking of humans in the field experiments involving PG.

The weights of the four models form the probability distribution over the values of the chance node, $Mod[M_j]$. As agent i is unaware of the true model of j, it may begin by assigning a uniform distribution to $Mod[M_j]$. Over time, this distribution is updated to reflect any information that i may have about j's model.

3.2. Bayesian Model Identification in I-IDs

As we mentioned before, i hypothesizes a limited number of candidate models of its opponent j, $M_j = \{m_j^1,\ldots,m_j^p,\ldots,m_j^n\}$, and intends to ascertain the true model, m_j^*, of j in the course of interaction. On observing j's action, where the observation in round t is denoted by o_i^t, i may update the likelihoods (weights) of the candidate models in the model node of the I-ID. Gradually, the model that emerges as most likely may be hypothesized to be the true model of j. Here, we explore the traditional setting, $m_j^* \in M_j$ where the true model, m_j^*, is in the model space, M_j, and move on to the challenge where the true model is outside it, $m_j^* \notin M_j$, in Section 4.

Let $o_i^{1:t-1}$ be the history of agent i's observations up to time $t-1$. Agent i's belief over the models of j at time step $t-1$ may be written as, $Pr(M_j|o_i^{1:t-1}) \stackrel{def}{=} \langle$

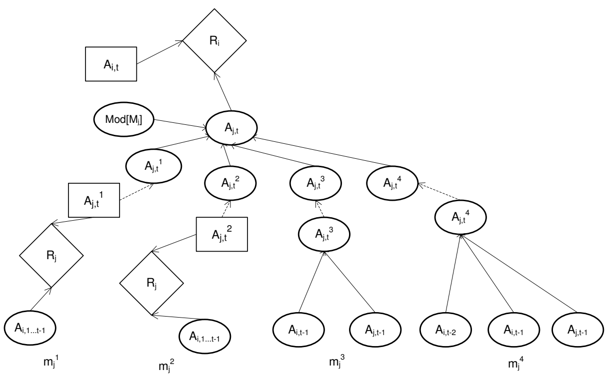


Fig. 3. Example level 1 I-ID for the repeated PG game with four models ascribed to j. The dashed arrows represent the mapping between decision or chance nodes in j's models and chance nodes in the model node.

$Pr(m_j^1|o_i^{1:t-1}), Pr(m_j^2|o_i^{1:t-1}),\ldots,Pr(m_j^*|o_i^{1:t-1}),\ldots, Pr(m_j^n|o_i^{1:t-1})\ \rangle$. If o_i^t is the observation at time t, agent i may update its belief on receiving the observation using a straightforward Bayesian process. We show the update of the belief over some model, m_j^n:

$$Pr(m_j^n|o_i^{1:t}) = \frac{Pr(o_i^t|m_j^n, o_i^{1:t-1})Pr(m_j^n|o_i^{1:t-1})}{\sum_{m_j \in M_j} Pr(o_i^t|m_j, o_i^{1:t-1})Pr(m_j)} \quad (1)$$

Here, $Pr(o_i^t|m_j^n, o_i^{1:t-1})$ is the probability of j performing the observed action given that its model is m_j^n. This may be obtained from the chance node A_j^n in the I-ID of i.

Eq. 1 provides a way for updating the weights of models contained in the model node, $Mod[M_j]$, given the observation history. In the context of the I-ID, agent i's belief over the other's models updated using the process outlined in Eq. 1 will converge in the limit. Formally,

Proposition 1 (Bayesian Learning in I-IDs)**.** *If an agent's prior belief assigns a non-zero probability to the true model of the other agent, its posterior beliefs updated using Bayesian learning will converge with probability 1.*

Proof of Proposition 1 relies on showing that the sequence of the agent's beliefs updated using Bayesian learning is known to be a Martingale [5]. Proposition 1 then follows from a straightforward application of the Martingale convergence theorem (§4 of Chapter 7 in Doob [5]). Doshi and Gmytrasiewicz [6] provide more details about this proof.

The above result does not imply that an agent's belief always converges to the true model of the other agent. This is due to the possible presence of models of the other agent that are *observationally equivalent* to the true model. Agent j's models that induce identical distributions over all possible future observation paths are said to be observationally equivalent for agent i. When a particular observation history obtains, agent i is unable to distinguish between the observationally equivalent models of j. In other words, the observationally equivalent models generate distinct behaviors for histories which are never observed.

Example: For an example of observationally equivalent models, consider the PG game introduced previously. Let agent i consider two candidate models of j. Suppose that as a best response to its belief, one of j's models leads to a strategy in which it would select *FC* for an infinite number of steps, but if at any time i chooses *D*, j would also do so at the next time step and then continue with *D*. The other model of j adopts a tit-for-tat strategy, i.e. j performs the action which i did in the previous time step. If agent i decides to select *FC* an infinite number of times, then the two models of j are observationally equivalent. Given i's strategy, both the candidate models of j assign a probability 1 to the observation history $\{ \langle$ *FC,FC* $\rangle, \langle$ *FC,FC* $\rangle, \ldots\}$, although the strategies are distinct.

4. Information-theoretic Model Identification

For practical purposes, the space of candidate models ascribed to j is often bounded. In the absence of prior knowledge, i may be unaware whether j's true model, m_j^*, is within the model space. If $m_j^* \notin M_j$ and in the absence of observationally equivalent models, Bayesian learning may be inadequate ($Pr(o_i^t|m_j^n, o_i^{1:t-1})$ in Eq. 1 may be 0 for all m_j^n). As bounded expansions of the model space do not guarantee inclusion of the true model, we seek to find a candidate model or a combination of models from the space, whose predictions are *relevant* in determining actions of j.

4.1. Relevant Models and Mutual Information

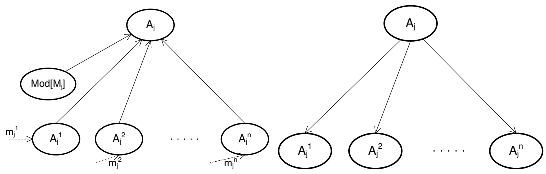
As the true model may lie outside the model space, our objective is to identify candidate models whose predictions exhibit a mutual pattern with the observed actions of the other agent. We interpret the existence of a mutual pattern as evidence that the candidate model shares some behavioral aspects with the true model. In order to do this, we introduce a notion of *relevance* between a model in M_j and the true model, m_j^*.

Let a_j^* be the observed action of the other agent j and $\bar{a}_j^*$ denote any other action from its set of actions. Define $Pr_{m_j^n}(a_j^n|a_j^*)$ as the probability that a candidate model of j, m_j^n, predicts action a_j^n when a_j^* is observed in the same time step.

Definition 1 (Relevant Model). *If for a model, m_j^n, and some observed action, a_j^*, there exists an action, a_j^n: $Pr_{m_j^n}(a_j^n|a_j^*) > Pr_{m_j^n}(a_j^n|\bar{a}_j^*)$, for all $\bar{a}_j^*$, where $a_j^n \in OPT(m_j^n)$ and the subscript m_j^n denotes the generative model, then m_j^n is a* relevant *model.*

Definition 1 formalizes the intuition that a relevant model predicts an action that is likely to correlate with a particular observed action of the other agent. In predicting a_j^n, model m_j^n may utilize the past observation history. We note that the above definition generalizes to a relevant combination of models in a straightforward way. Given Def. 1, we need an approach that assigns large probabilities to the relevant model(s) in the node $Mod[M_j]$ over time. We proceed to show one way of computing these probabilities.

We begin by observing that the chance nodes, $Mod[M_j]$, A_j and the mapped chance nodes, $A_j^1, A_j^2, \ldots$, form a BN, as shown in Fig. 4(a). We seek the weights of models in $Mod[M_j]$ that would allow the distribution over A_j to resemble that of the observed actions. Subsequently, we may map the problem to one of classifying the predicted actions of the individual models with the observed action of j, and using the classification function for deriving the model weights. Because the candidate models are independent of each other, the BN is *naive* and the classification reduces to learning the parameters (CPDs) of the naive BN using say, the maximum likelihood approach with Dirichlet priors. For multiple agents, the models may exhibit dependencies in which case we learn a general BN. We show the equivalent naive BN in Fig. 4(b).



(a) (b)

Time	A_j^1	A_j^2	...	A_j^n	A_j
1	PC	D	...	D	PC
2	D	PC	...	FC	FC
3	PC	PC	...	D	PC
4	FC	FC	...	PC	D
5	D	PC	...	PC	FC
6	FC	FC	...	D	FC
.	...	...	...	...	...

(c)

Fig. 4. (a) The BN in the I-ID of agent i; (b) The equivalent naive BN that we use for classifying the outcomes of the candidate models to the observation history; (c) Example of the training set used for learning the naive BN for PG. The actions in column A_j are observations of i, while remaining columns are obtained from models.

As relevant models hint at possible dependencies with the true model in terms of predicted and observed actions, we utilize the *mutual information* (MI) [4] between the chance nodes A_j and say, A_j^n, as a measure of the likelihood of the model, m_j^n, in $Mod[M_j]$. MI is a well-known way of quantifying the mutual dependency between two random variables.

Definition 2 (Mutual Information)**.** *The mutual information (MI) of the true model, m_j^* and a candidate model, m_j^n, is computed as:*

$$\begin{aligned} MI(m_j^n, m_j^*) &\stackrel{def}{=} Pr(A_j^n, A_j) log[\frac{Pr(A_j^n, A_j)}{Pr(A_j^n)Pr(A_j)}] \\ &= Pr(A_j^n|A_j) Pr(A_j) log[\frac{Pr(A_j^n|A_j)}{Pr(A_j^n)}] \end{aligned} \quad (2)$$

Here, A_j^n is the chance node mapped from the model, m_j^n, and A_j is the chance node for the observed actions generated by the true model, m_j^*.

The terms $Pr(A_j^n|A_j)$, $Pr(A_j^n)$ and $Pr(A_j)$ are calculated from the CPDs of the naive BN. Note that the distribution, $Pr(A_j^n|A_j)$, implies possible relations between observed and predicted actions in the history. Here, the observed history of j's actions together with the predictions of the models over time may serve as the training set for learning the parameters of the naive BN. We show an example training set for PG in Fig. 4(c). Values of the columns, A_j^1, A_j^2, ..., A_j^n are obtained by solving the corresponding models and sampling the resulting distributions if needed. We utilize the normalized MI at each time step as the model weights in the chance node, $Mod[M_j]$.

Example: We show an example training set for PG in Fig. 4(c). Notice that:

$$Pr_{m_j^1}(A_j^1 = PC|A_j = PC) = \frac{Pr(A_j^1=PC, A_j=PC)}{Pr(A_j=PC)} = \frac{2/6}{2/6} = 1$$
$$Pr_{m_j^1}(A_j^1 = PC|A_j = FC) = \frac{Pr(A_j^1=PC, A_j=FC)}{Pr(A_j=FC)} = \frac{0/6}{3/6} = 0$$
$$Pr_{m_j^1}(A_j^1 = PC|A_j = D) = \frac{Pr(A_j^1=PC, A_j=D)}{Pr(A_j=D)} = \frac{0/6}{1/6} = 0$$

Therefore, we get $Pr_{m_j^1}(A_j^1 = PC|A_j = PC) > Pr_{m_j^1}(A_j^1 = PC|A_j = FC$ or $D)$. Hence, from Definition 1 we conclude that the model m_j^1 that maps to A_j^1 is a relevant model so far. Additionally,

$$\begin{aligned} &MI(m_j^1, m_j^*) = Pr(A_j^1|A_j)Pr(A_j) log\frac{Pr(A_j^1|A_j)}{Pr(A_j^1)} \\ &= \sum_{a_j}\sum_{a_j^1 \in OPT(m_j^1)}[Pr_{m_j^1}(a_j^1|a_j)Pr(a_j) \\ &log\frac{Pr_{m_j^1}(a_j^1|a_j)}{Pr_{m_j^1}(a_j^1)}] \\ &= 1 \cdot log[\frac{1}{\frac{2}{6}\cdot\frac{2}{6}}] + \frac{2}{6}\cdot log[\frac{\frac{2}{6}}{\frac{3}{6}\cdot\frac{2}{6}}] + 1 \cdot log[\frac{1}{\frac{3}{6}\cdot\frac{2}{6}}] \\ &+ \frac{1}{6}\cdot log[\frac{\frac{1}{6}}{\frac{1}{6}\cdot\frac{2}{6}}] = 0.551 \ \ (after\ \ normalization) \end{aligned}$$

4.2. Theoretical Results

Obviously, model m_j^n is irrelevant if $Pr_{m_j^n}(a_j^n|a_j^*) = Pr_{m_j^n}(a_j^n|\bar{a}_j^*)$ for each $a_j^n \in OPT(m_j^n)$ and all $\bar{a}_j^*$. Then, we trivially obtain the next proposition.

Proposition 2. *If m_j^n is irrelevant, $MI(m_j^n, m_j^*) = 0$.*

Proof. We may express $MI(m_j^n, m_j^*)$ in terms of $Pr_{m_j^n}(a_j^n|a_j^*)$ and $Pr_{m_j^n}(a_j^n|\bar{a}_j^*)$ as below:

$$\begin{aligned} MI(m_j^n, m_j^*) = \sum_{a_j^1 \in OPT(m_j^n)} \Big\{ & Pr_{m_j^n}(a_j^n|a_j^*)Pr(a_j^*) \\ & log[\frac{Pr_{m_j^n}(a_j^n|a_j^*)}{Pr_{m_j^n}(a_j^n|a_j^*)Pr(a_j^*) + Pr_{m_j^n}(a_j^n|\bar{a}_j^*)Pr(\bar{a}_j^*)}] \\ & + Pr_{m_j^n}(a_j^n|\bar{a}_j^*)Pr(\bar{a}_j^*) \\ & log[\frac{Pr_{m_j^n}(a_j^n|\bar{a}_j^*)}{Pr_{m_j^n}(a_j^n|a_j^*)Pr(a_j^*) + Pr_{m_j^n}(a_j^1|\bar{a}_j^*)Pr(\bar{a}_j^*)}] \Big\} \end{aligned} \quad (3)$$

Since $Pr_{m_j^n}(a_j^n|a_j^*) = Pr_{m_j^n}(a_j^n|\bar{a}_j^*)$, we have $Pr_{m_j^n}(a_j^n|a_j^*) Pr(a_j^*) + Pr_{m_j^n}(a_j^n|\bar{a}_j^*)Pr(\bar{a}_j^*)$=$Pr_{m_j^n}(a_j^n|a_j^*)$= $Pr_{m_j^n}(a_j^n|\bar{a}_j^*)$. Consequently, the $log(\cdot)$ term in Eq. 3 becomes zero, which leads to $MI(m_j^n, m_j^*) = 0$. ∎

As MI is non-negative, Proposition 2 implies that relevant models are assigned a higher MI than irrelevant ones. To enable further analysis, we compare the relevance among candidate models.

Definition 3 (Relevance Ordering)**.** *Let a_j^* be some observed action of the other agent j. If for two relevant models, such that $Pr_{m_j^n}(a_j^n|a_j^*) > Pr_{m_j^p}(a_j^p|a_j^*)$ and $Pr_{m_j^n}(a_j^n|\bar{a}_j^*) < Pr_{m_j^p}(a_j^p|\bar{a}_j^*)$, for all $\bar{a}_j^*$ where $a_j^n \in OPT(m_j^n)$, $a_j^p \in OPT(m_j^p)$, and $\bar{a}_j^*$ denotes any other action of the true model, then m_j^n is a* more relevant *model than m_j^p.*

Given Def. 3, we show that models which are *more relevant* are assigned a higher MI. Proposition 3 formalizes this observation. The proof below adapts [13].

Proposition 3. *If m_j^n is a more relevant model than m_j^p as per Definition 3 and m_j^* is the true model, then $MI\ (m_j^n, m_j^*) > MI\ (m_j^p, m_j^*)$.*

Proof. We further expand Eq. 3 and express $MI(m_j^n, m_j^*)$ as below:

$$MI(m_j^n, m_j^*) = \sum_{a_j^n \in OPT(m_j^n)} \Big\{ Pr_{m_j^n}(a_j^n|a_j^*) Pr(a_j^*) \\ log[\frac{1}{Pr(a_j^*) + \frac{Pr_{m_j^n}(a_j^n|\bar{a}_j^*)}{Pr_{m_j^n}(a_j^n|a_j^*)} Pr(\bar{a}_j^*)}] \\ + Pr_{m_j^n}(a_j^n|\bar{a}_j^*) Pr(\bar{a}_j^*) \\ log[\frac{1}{\frac{Pr_{m_j^n}(a_j^n|a_j^*)}{Pr_{m_j^n}(a_j^n|\bar{a}_j^*)} Pr(a_j^*) + Pr(\bar{a}_j^*)}] \Big\} \quad (4)$$

Notice that $Pr_{m_j^n}(a_j^n|a_j^*) > Pr_{m_j^n}(a_j^n|\bar{a}_j^*)$, we get $Pr(a_j^*) + \frac{Pr_{m_j^n}(a_j^n|\bar{a}_j^*)}{Pr_{m_j^n}(a_j^n|a_j^*)} Pr(\bar{a}_j^*) < 1$ since $Pr(a_j^*) + \frac{Pr_{m_j^n}(a_j^n|\bar{a}_j^*)}{Pr_{m_j^n}(a_j^n|\bar{a}_j^*)} Pr(\bar{a}_j^*) = 1$, similarly $\frac{Pr_{m_j^n}(a_j^n|a_j^*)}{Pr_{m_j^n}(a_j^n|\bar{a}_j^*)} Pr(a_j^*) + Pr(\bar{a}_j^*) > 1$. Hence, in Eq. 4, the first term, $Pr_{m_j^n}(a_j^n|a_j^*)$ $Pr(a_j^*)$ $log[\frac{1}{Pr(a_j^*) + \frac{Pr_{m_j^n}(a_j^n|\bar{a}_j^*)}{Pr_{m_j^n}(a_j^n|a_j^*)} Pr(\bar{a}_j^*)}] > 0$ (since the base is 2 in log defined in MI), while the second term, $Pr_{m_j^n}(a_j^n|\bar{a}_j^*)$ $Pr(\bar{a}_j^*)$ $log[\frac{1}{\frac{Pr_{m_j^n}(a_j^n|a_j^*)}{Pr_{m_j^n}(a_j^n|\bar{a}_j^*)} Pr(a_j^*) + Pr(\bar{a}_j^*)}] < 0$.

Then, for a fixed $Pr(a_j^*)$, $MI(m_j^n, m_j^*)$ is a monotonically increasing function of $Pr_{m_j^n}(a_j^n|a_j^*)$ for a fixed $Pr_{m_j^n}(a_j^n|\bar{a}_j^*)$, and a monotonically decreasing function of $Pr_{m_j^n}(a_j^n|\bar{a}_j^*)$ for a fixed $Pr_{m_j^n}(a_j^n|a_j^*)$ since the second term is less than zero in Eq. 4. Therefore, substituting $Pr_{m_j^n}(a_j^n|a_j^*)$ and $Pr_{m_j^n}(a_j^n|\bar{a}_j^*)$ with $Pr_{m_j^p}(a_j^p|a_j^*)$ ($< Pr_{m_j^n}(a_j^p|a_j^*)$) and $Pr_{m_j^p}(a_j^p|\bar{a}_j^*)$ ($> Pr_{m_j^n}(a_j^n|\bar{a}_j^*)$) respectively, results in $MI(m_j^n, m_j^*) > MI(m_j^p, m_j^*)$. ∎

For the sake of completeness, we show that if the true model, m_j^*, is contained in the model space, our approach analogous to Bayesian learning will converge.

Proposition 4 (Convergence)**.** *Given that the true model $m_j^* \in M_j$ and is assigned a non-zero probability, the normalized distribution of mutual information of the models converges with probability 1.*

Proof. The proof is intuitive and relies on the fact that the estimated parameters of the naive Bayes converge to the true parameters as the observation history grows (see chapter 3 of Rennie [16] for the proof when the *maximum a posteriori* approach is used for parameter estimation). Proposition 4 then follows because the terms $Pr(A_j^n|A_j)$, $Pr(A_j^n)$ and $Pr(A_j)$ used in calculating the MI are obtained from the parameter estimates of the naive BN. ∎

Analogous to Bayesian learning, the distribution of MI may not converge to the true model in the presence of *MI-equivalent* models in M_j. In particular, the set of MI-equivalent models is larger and includes observationally equivalent models. However, consider the example where j's true strategy is to always select *FC*, and let M_j include the true model and a candidate model that generates the strategy of always selecting *D*. Though observationally distinct, the two candidate models are assigned equal MI due to the perceived dependency between the action of selecting *D* by the candidate and selecting *FC* by the true one. However, in node A_j, the action *D* is classified to the observed, *FC*.

Model Weight Update
Input: I-ID of agent i, observation o_i^t, training set Tr

1. Agent i receives an observation o_i^t
2. Solve the models, m_j^p ($p = 1, \ldots, n$) to get actions for the chance nodes A_j^p ($p = 1, \cdots, n$)
3. Add ($A_j^1, \cdots, A_j^p, \cdots, A_j^n, o_i^t$) as a sample into the training set *Tr*
4. Learn the parameters of the *naive BN* including the chance nodes, $A_j^1, \ldots, A_j^n$, and A_j
5. **For each** A_j^p ($p = 1, \ldots, n$) **do**
6. Compute $MI(m_j^p, m_j^*)$ using Eq. 2
7. Obtain $Pr(A_j|A_j^p)$ from the learned *naive BN*
8. Populate CPD of the chance node A_j in the I-ID using $Pr_{m_j^p}(A_j|A_j^p)$
9. Normalize $MI(m_j^p, m_j^*)$
10. Populate CPD of the chance node $Mod[M_j]$ using MI

Fig. 5. Algorithm revises the model weights in the model node, $Mod[M_j]$, on observing j's action using MI as a measure of likelihood, and populates CPDs of the chance node, A_j, using the learned naive BN.

4.3. *Algorithm*

We briefly outline the algorithm for model identification in Fig. 5. In each round t, agent i receives an observation of its opponent j's action (line 1). This observation together with solutions from candidate models of j (line 2), constitute one sample in the training set Tr (line 3; see Fig. 4(c)). The training set is used for learning the parameters of the naive BN (line 4) and subsequently for computing the model weights in the I-ID. Given the learned parameters, we compute the MI of each candidate model m_j^p and m_j^* (line 6). The posterior probabilities (from line 7) are also used in the CPD of the chance node A_j in the I-ID (line 8). Notice that the CPD, $Pr_{m_j^p}(A_j|A_j^p)$, describes the relation between the predicted actions by candidate models and the observed actions. In other words, it reflects the classification of the predicted actions. The normalized MI is assigned as the distribution of the chance node $Mod[M_j]$ in the I-ID (line 10). This distribution represents the updated weight over the candidate models of j. Given the updated model weights and the populated CPDs of the chance node A_j, we solve the I-ID of agent i to obtain its action.

5. Performance Evaluation

We evaluate the effectiveness of the algorithm outlined in Fig. 5 in the context of three well-known repeated games: the repeated PG game, repeated one-shot negotiations as in [17] though simplified, and repeated Rock-Paper-Scissor games. As we mentioned previously, if the true model falls outside the model space ($m_j^* \notin M_j$), Bayesian learning (BL) may be inadequate. A simple adaptation of BL (A-BL) would be to restart the BL process when the likelihoods become zero by assigning candidate models prior weights using the frequency with which the observed action has been predicted by the candidate models so far. Additionally, we utilize another information-theoretic measure, the KL-Divergence (KL), which is a well-known pseudo-distance measure between two probability distributions, to assign the likelihood of a candidate model. Lower is the KL between distributions over A_j^n and A_j, larger is the likelihood of the corresponding model, m_j^n.

We let agents i and j play 1000 rounds of each game and report i's average rewards. To facilitate analysis, we also show the changing model weights across rounds that are assigned to the relevant and true models for both cases: $m_j^* \in M_j$ and $m_j^* \notin M_j$.

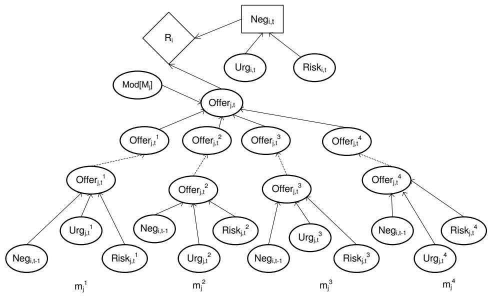
5.1. Repeated Public Good Game

In the PG game, we utilize the I-ID in Fig. 3 to model the interaction. Agent i plays with the opponent j multiple rounds of PG and aims to gain larger rewards in the long run by discovering j's true behavioral model. For the setting, $m_j^* \in M_j$, we let the model space, M_j, contain three models, m_j^1, m_j^3, and m_j^4, and let agent j play using the true model, m_j^4. Fig. 6(a) demonstrates the favorable performances of MI, BL and A-BL, which quickly converge to the true model and gain almost the same average rewards. Note that KL assigns non-zero weights to other models as the distribution generated by those candidates is somewhat close to that of the true model.

For evaluation of the case where $m_j^* \notin M_j$, i considers three candidate models of j, m_j^2, m_j^3, and m_j^4, while j uses the reciprocal model m_j^1. We observe that MI significantly outperforms other updating methods obtaining the largest average rewards over the long run (Fig. 6(b)). This is because MI finds the deliberative model, m_j^4, to be most relevant to the true model, m_j^1. Model m_j^1 expects i to perform its most frequently observed action and matches it, an aspect that is best shared by m_j^4, which relies the most on other's actions. We note that MI does not monotonically increase but assigns the largest weight to the most relevant model at any point in time. Notice that both m_j^1 and m_j^4 consider actions of the other agent, and identical actions of the agents as promoted by a reciprocal model are more valuable. Both the A-BL and KL methods settle on the altruistic model, m_j^2, as the most likely.

j, i	A	R	C
Low	-1	1	1
Mid	1	0	1
High	3	1	-1

(a)



(b)

Fig. 7. (a) Single shot play of a negotiation between the seller i and buyer j. The numbers represent the payoffs of the seller i. (b) I-ID for the seller in the negotiation with four models ascribed to the buyer j.

5.2. Repeated One-shot Negotiations

A seller agent i wants to sell an item to a buyer agent j. The buyer agent bargains with the seller and offers a price that ranges from Low, Mid, to $High$. The seller agent decides whether to $accept$ the offer (A), to $reject$ it immediately (R), or to $counter$ the offer (C). If i counters the offer, it expects a new price offer from agent j. Once the negotiation is completed

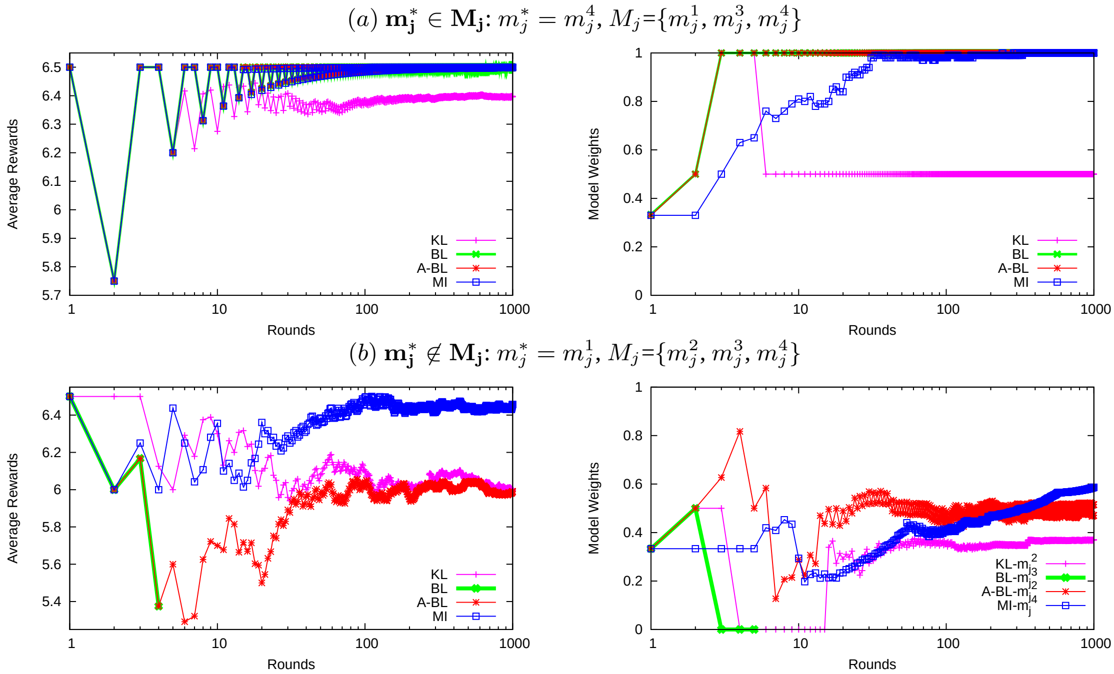


Fig. 6. Performance profiles for both, the traditional setting, $m_j^* \in M_j$, and the realistic case, $m_j^* \notin M_j$, in the repeated PG game. Notice that, for the case of $m_j^* \notin M_j$, the model weight assigned using BL drops to zero.

successfully or fails, the agents restart a new one on a different item; otherwise, they continue to bargain. Figure 7(a) shows the payoffs of the seller agent when interacting with the buyer. The seller aims to profit by getting large rewards (payoff) in the bargaining process. As in most cases of negotiations, here the seller and the buyer are unwilling to share their preferences with the other. For example, from the perspective of the seller, some types of buyer agents have different bargaining strategies based on their risk preferences. The ability to identify the buyer's true model enables the seller agent to choose rational actions in the negotiation.

The idea of using probabilistic graphical models in multiagent negotiation was previously explored in [15]. In a similar vein, we model agent i using the I-ID shown in Fig. 7(b). Analogous to [17], we consider four types of the buyer agent j. Each of them is represented using a BN. They differ in the probability distributions for the chance nodes *Risk* that represents the buyer's risk attitude and *Urg*, which represents the urgency of the situation to the agent. Let model m_j^1 represent a buyer of a *risk averse* type. A risk averse agent has an aversion to losing the deal and hence always proposes a high offer. The second model, m_j^2, is a *risk seeking* buyer that adopts a risky strategy by intending to offer a low price. Model m_j^3 is a *risk neutral* buyer that balances its low and high offers in the negotiation. The final model, m_j^4, is a buyer that is risk neutral but in an urgent situation, and is eager to acquire the item. Consequently, it is prone to offering a high price, though its actions also depend on the seller. Note that the chance node $Neg_{i,t-1}$ represents i's previous action in the negotiation.

Let agent i consider three candidate models for j, m_j^1, m_j^2, and m_j^3, and agent j uses model m_j^1 for the setting, $m_j^* \in M_j$. Fig. 8(a) reveals that all the different updating methods correctly identify the true model after some steps and gather similar rewards. As j is risk averse, it often offers a high price that the seller chooses to accept incurring a payoff of 3.

In the case where $m_j^* \notin M_j$, agent j plays the game using the model, m_j^4, and i assumes the remaining three models as candidates. Notice that MI eventually assigns the largest weight ($\approx$ 0.63) to the risk averse agent, m_j^1, that always offers a high price in the negotiation. This behavior is consistent with the model, m_j^4, that represents an urgent buyer who is also prone to offering a high price. Consequently, MI obtains better average rewards than other methods. The remaining two candidate models are MI-equivalent. In comparison, both KL and A-BL methods eventually identify

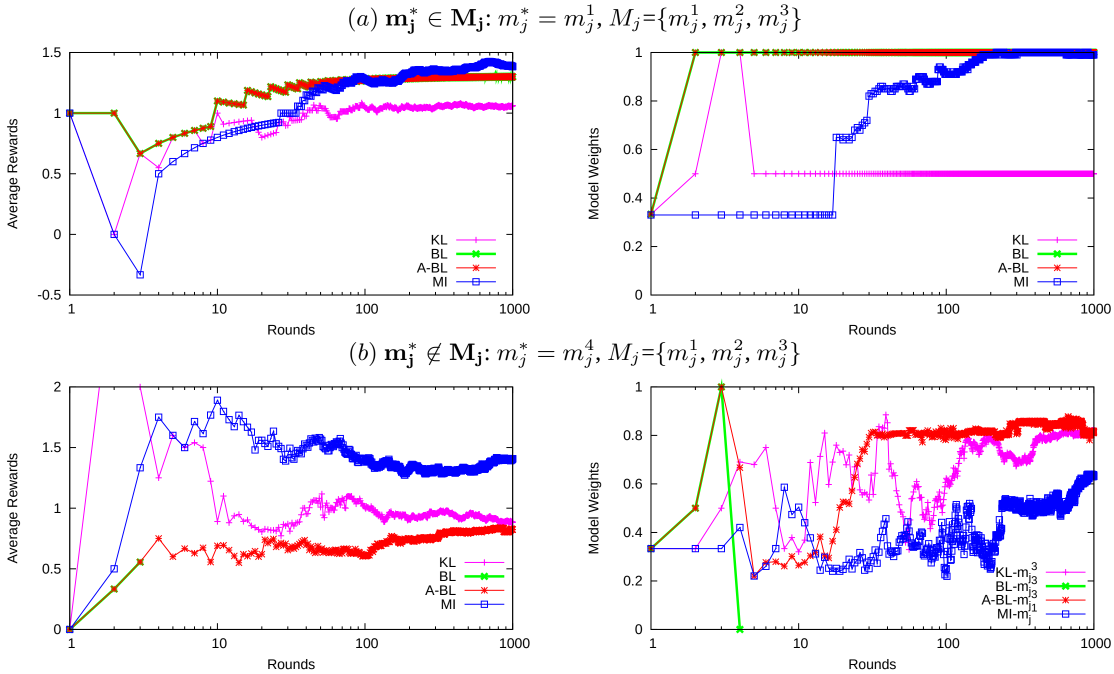


Fig. 8. Performance profiles and the changing model weights for the two cases while repeatedly playing the negotiation game.

the risk neutral agent m_j^3, which leads to lower average rewards.

5.3. *Rock-Paper-Scissor Games*

Two agents, i and j, play Rock-Paper-Scissor (RPS; also called RoShamBo) game repeatedly a finite number of times and aim for winning the most number of times thereby gathering larger average rewards (payoff). After each round, only the simultaneous actions of agents are exhibited to each other. The payoffs in the game are as shown in Table 2.

Table 2
Payoff for agents, i and j, in a RPS game.

i, j	Rock	Paper	Scissor
Rock	(0,0)	(-1,1)	(1,-1)
Paper	(1,-1)	(0,0)	(-1,1)
Scissor	(-1,1)	(1,-1)	(0,0)

We model four types of agent j from the perspective of agent i in the experiment. The I-ID in Fig. 9 shows that agent i considers four models of j (m_j^1, m_j^2, m_j^3, and m_j^4) in the model node at time t. The first model, m_j^1, is a simple ID where the chance node $A_{i,\langle 1,\cdots,t-1\rangle}$ models the frequencies of agent i's different actions in the history (from 1 to time $t-1$).

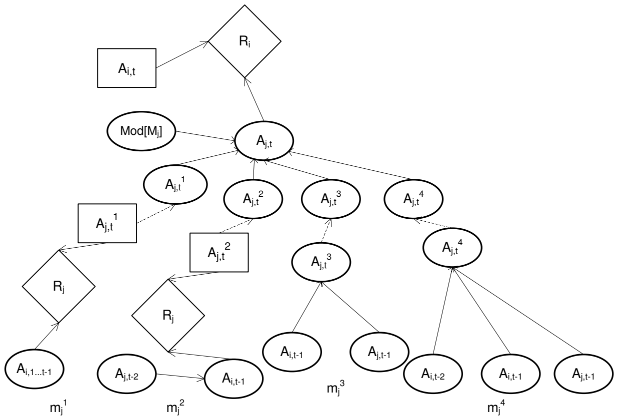


Fig. 9. Example I-ID for the Rock-Paper-Scissor game with four models ascribed to j.

The second ID, m_j^2, has a different structure: Agent j's belief over i's actions depends on j's behaviors in the previous time step $t-1$. Thus agent j thinks that agent i may play according to what j plays in the previous time step. The remaining two models are BNs that reflect j's more deliberative behaviors. The third model, m_j^3, represents j's behavior of counting both its own actions in the previous time step ($A_{j,t-1}$) and i's previous actions ($A_{i,t-1}$). The final model m_j^4 shows j has a more sophisticated decision process. Not only agents i and j's behaviors at time $t-1$ (chance nodes

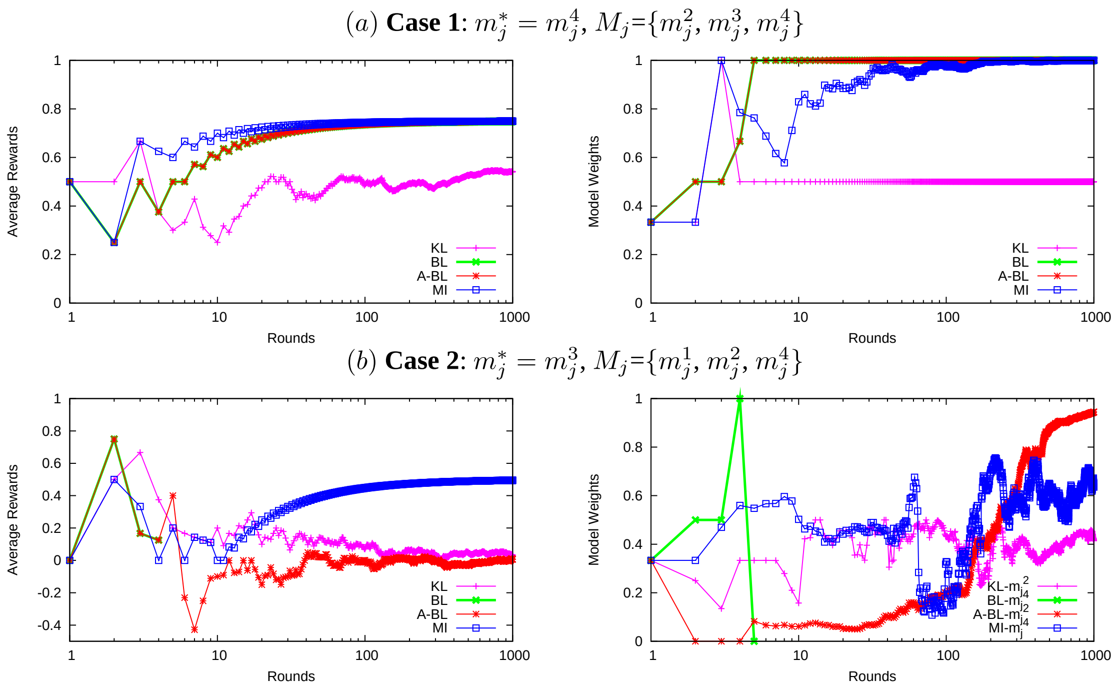


Fig. 10. Performance profiles and revised model weights for the two cases for the RPS game.

$A_{i,t-1}$ and $A_{j,t-1}$) but also agent i's actions at time $t-2$ ($A_{i,t-2}$) are considered when agent j decides its behaviors at time t. Note that the strategies reflected in these four models are often used in competitive RPS tournaments.

We assume that for case 1, agent j plays the true model, m_j^4, and i uses a model space consisting of the three models, $\{m_j^2, m_j^3, m_j^4\}$. In Fig. $10(a)$, we show the average rewards obtained by i as well as the varying model weights (assigned to m_j^4). As we may expect, both the BL and A-BL methods quickly identify the true model and therefore gather large average rewards. The MI method also identifies m_j^4 among the candidate models after a few more steps, and gradually gains identical rewards as BL from play. Notice that KL does not perform as well because candidate models other than m_j^4 are assigned non-zero model weights. However, it does assign the largest likelihood to m_j^4.

For case 2 (Fig. $10(b)$), we let agent j use the model m_j^3 while i hypothesizes the remaining three models for j. We first observe that MI gains the largest average reward. MI eventually identifies the model, m_j^4, as the most relevant. This is intuitive because, analogous to the true model m_j^3, the model m_j^4 also deliberates actions based on both opponents' and its own previous behaviors. BL filters out all candidate models quickly due to contradicting observations (notice the drop to zero for m_j^4). Both A-BL and KL find m_j^2 as the most likely.

6. Discussion

I-IDs use Bayesian learning to update beliefs with the implicit assumption that true models of other agents are contained in the model space. As model spaces are often bounded in practice, true models of others may not be present in the space. We show that distribution of MI of the candidate models learned by classifying their predictions exhibits a performance comparable to Bayesian learning when the true model is within the set of candidate models. More importantly, the MI approach improves on other heuristic approaches for the plausible case that true model is outside the model space. Thus, the approach shows potential as a general purpose candidate technique for identifying models when we are uncertain whether the model space is exhaustive. However, an important limitation is that the space of MI-equivalent models is large. While it may not affect performance, it merits further investigation.

Acknowledgment

Yifeng Zeng acknowledges support from both the Obel Family Foundation in Denmark and National Natural Science Foundation of China (No. 60974089 and No. 60975052). Prashant Doshi is supported in part by AFOSR through grant no. #FA9550-08-1-0429 and in part by a NSF CAREER award with grant no. #IIS-0845036. Views and findings expressed in this article are those of the authors alone and do not reflect in any way upon the funding agencies. The authors also wish to thank the anonymous reviewers for their valuable comments.

References

[1] N. Bard and M. Bowling. Particle filtering for dynamic agent modelling in simplified poker. In *Proceedings of the Twenty-Second Conference on Artificial Intelligence (AAAI)*, pages 515–521, 2007.

[2] C. Camerer. *Behavioral Game Theory: Experiments in Strategic Interaction*. Princeton University Press, 2003.

[3] D. Carmel and S. Markovich. Learning models of intelligent agents. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI)*, pages 62–67, 1996.

[4] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2006.

[5] J. L. Doob. *Stochastic Processes*. John Wiley and Sons, 1953.

[6] P. Doshi and P. J. Gmytrasiewicz. On the difficulty of achieving equilibrium in interactive pomdps. In *Twenty First Conference on Artificial Intelligence (AAAI)*, pages 1131–1136, Boston, MA, 2006.

[7] P. Doshi, Y. Zeng, and Q. Chen. Graphical models for interactive pomdps: Representations and solutions. *Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS)*, 18(3):376–416, 2009.

[8] D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, 1991.

[9] Y. Gal and A. Pfeffer. Networks of influence diagrams: A formalism for reasoning about agents' decision-making processes. *Journal of Artificial Intelligence Research (JAIR)*, 33:109–147, 2008.

[10] P. J. Gmytrasiewicz, S. Noh, and T. Kellog. Bayesian update of recursive agent models. *International Journal of User Modeling and User-Adapted Interaction*, 8(1-2):49–69, 1998.

[11] H. Koen and T. Dmytro. Opponent modelling in automated multi-issue negotiation using bayesian learning. In *Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 331–338, 2008.

[12] D. Koller and B. Milch. Multi-agent influence diagrams for representing and solving games. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1027–1034, 2001.

[13] C.-L. Liu. Some theoretical properties of mutual information for student assessments in intelligent tutoring systems. In *Intl. Symp. on Foundations of Intelligent Systems*, 2005.

[14] N. S. Madsen and F. V. Jensen. An influence diagram framework for acting under influence by agents with unknown goals. In *Proceedings of the 4th European Workshop on Probabilistic Graphical Models (PGM)*, pages 289–296, 2008.

[15] C. Mudgal and J. Vassileva. An influence diagram model for multi-agent negotiation. In *International Conference on Multi-Agent Systems (ICMAS)*, pages 451–452, 2000.

[16] J. D. Rennie. Improving multi-text classification using naive bayes. Technical Report AI TR 2001-04, MIT, 2001.

[17] S. Saha, A. Biswas, and S. Sen. Modeling opponent decision in repeated one-shot negotiations. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 397–403, 2005.

[18] R. D. Shachter. Evaluating influence diagrams. *Operations Research*, 34(6):871–882, 1986.

[19] N. Sturtevant and M. Bowling. Robust game play against unknown opponents. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 713–719, 2006.

[20] D. Suryadi and P. Gmytrasiewicz. Learning models of other agents using influence diagrams. In *Proceedings of the Seventh International Conference on User Modeling (UM)*, pages 223–232, 1999.

[21] J. A. Tatman and R. D. Shachter. Dynamic programming and influence diagrams. *IEEE Transaction on Systems, Man, and Cybernetics*, 20(2):365–379, 1990.

[22] D. Zeng and K. Sycara. Bayesian learning in negotiation. *International Journal of Human Computer Systems*, 48:125–141, 1998.