An Information-Theoretic Approach to Model Identification in Interactive Influence Diagrams

Yifeng Zeng
Dept. of Computer Science
Aalborg University, Denmark
yfzeng@cs.aau.dk

Prashant Doshi
Dept. of Computer Science
University of Georgia, U.S.A.
pdoshi@cs.uga.edu

Abstract

Interactive influence diagrams (I-IDs) offer a transparent and semantically clear representation for the decision-making problem in multiagent settings. They ascribe procedural models such as IDs and I-IDs to the behavior of other agents. Procedural models offer the benefit of understanding how others arrive at their behaviors. However, as model spaces are often bounded, the true models of others may not be present in the model space. In addition to considering the case when the true model is within the model space, we investigate the realistic case when the true model may fall outside the 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 in two repeated games and provide results in support.*

1 Introduction

Interactive influence diagrams (I-IDs; Doshi, Zeng, & Chen [5]) are graphical models of decision-making in uncertain multiagent settings. I-IDs generalize influence diagrams (IDs; Tatman & Shachter [14]) to make them applicable to settings shared with other agents, who may act, observe and update their beliefs. I-IDs and their sequential counterparts, I-DIDs, contribute to a growing line of work that includes multiagent influence diagrams (MAIDs; Koller & Milch [8]), and more recently, networks of influence diagrams (NIDs; Gal & Pfeffer [7]). 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 (BNs), 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 as though there are uncountably infinite numbers of agent functions, there are only countable computable models. Hence, theoretically it is likely that an agent's true model may not be within the model space. This insight is not new; it motivated Suryadi and Gmytrasiewicz ([13]) 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 ([2]) learn finite state automatons to model agents' strategies, and Saha *et al.* ([11]) learn Chebychev polynomials to approximate agents' decision functions. However, the representations are non-procedural and the learning problems complex.

In this paper, we consider the realistic case that the true model may not be within the bounded model space 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 of 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 action. 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.

2 Background

We briefly describe interactive influence diagrams (I-IDs; Doshi, Zeng, & Chen [5]) for modeling two-agent interactions and illustrate their application using a simple example. We also discuss Bayesian learning in I-IDs for identifying models and point out a limitation.

2.1 Overview of Interactive Influence Diagrams

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. 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's actions, and a dashed link, called a *policy link*.

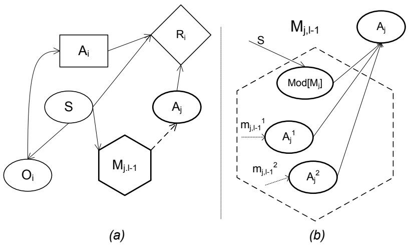


Figure 1. (a) Generic I-ID for agent i situated with one other agent j. The hexagon is the model node whose structure we show in (b). Members of model node may be IDs, BNs or I-IDs themselves (m_j^1, m_j^2; not shown here for simplicity) whose decision nodes are mapped to the corresponding chance nodes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

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 $m_{j,l-1}$ within an I-ID. 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. 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 is given 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 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 an I-ID proceeds in a bottom-up manner, and is implemented recursively. We start by solving the lower level models, which are traditional IDs or BNs. 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. 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, $Mod[M_j]$. The transformed I-ID is a traditional ID that may be solved using the standard expected utility maximization method [12].

Illustration We illustrate I-IDs using an example application to the public good (PG) game with punishment (Table 1) explained in detail in [6]. 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) without making 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

[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.

i, j	FC	PC	D
FC	$(2c_i X_T,2c_j X_T)$	$(\frac{3}{2}X_T c_i - \frac{1}{2}c_p,\frac{1}{2}X_T + \frac{3}{2}X_T c_j - \frac{1}{2}P)$	$(c_i X_T - c_p,X_T + c_j X_T - P)$
PC	$(\frac{1}{2}X_T + \frac{3}{2}X_T c_i - \frac{1}{2}P,\frac{3}{2}X_T c_j - \frac{1}{2}c_p)$	$(\frac{1}{2}X_T + c_i X_T,\frac{1}{2}X_T + c_j X_T)$	$(\frac{1}{2}X_T + \frac{1}{2}c_i X_T - \frac{1}{2}P,X_T + \frac{1}{2}c_j X_T - P)$
D	$(X_T + c_i X_T - P,c_j X_T - c_p)$	$(X_T + \frac{1}{2}c_i X_T - P,\frac{1}{2}X_T + \frac{1}{2}c_j X_T - \frac{1}{2}P)$	(X_T,X_T)

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

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 larger average rewards. After a round, 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 [1], different types of agents play PG differently. To act rationally, i ascribes candidate behavioral models to j. We assume the models are procedural taking the form of IDs and BNs.

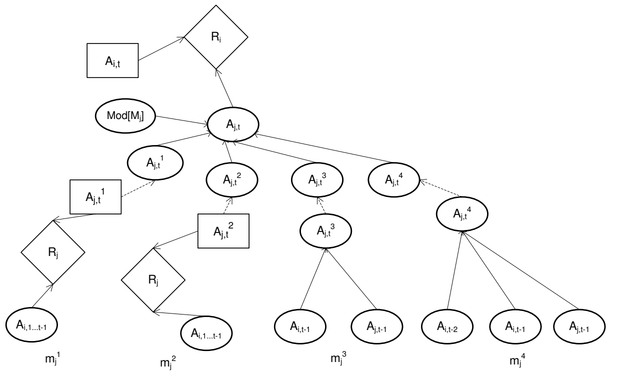


Figure 2. Example level 1 I-ID for the repeated PG game with four models ascribed to j. The dashed arrows represent the mapping between decision or chance nodes in j's models and chance nodes in the model node.

For illustration, let agent i consider four models of j (m_j^1, m_j^2, m_j^3, and m_j^4) in the model node at time t, as shown in Fig. 2. The first two models, m_j^1 and m_j^2, 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_j^1 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 as well. The model m_j^2 has a high c_j and represents an altruistic agent who prefers to contribute during the play. The third model, m_j^3, 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.

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.

2.2 Bayesian Model Identification in I-IDs

As we mentioned before, i hypothesizes a limited number of candidate models of its opponent j, $M_j = \{m_j^1,..,m_j^p, \ldots,m_j^n\}$, 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 3.

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, $Pr(M_j|o_i^{1:t-1}) \overset{def}{=} \langle\, Pr(m_j^1)$, $Pr(m_j^2)$,..,$Pr(m_j^*)$,.., $Pr(m_j^n)\,\rangle$. 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, in Eq. 1.

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

Here, $Pr(o_i^t|m_j^n)$ 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 [4]. Proposition 1 then follows from a straightforward application of the Martingale convergence theorem (§4 of Chapter 7 in Doob [4]).

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. The observationally equivalent models generate distinct behaviors for histories which are never observed.

3 Information-Theoretic Model Identification in I-IDs

For computability 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 ($Pr(o_i^t|m_j^n)$ 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.

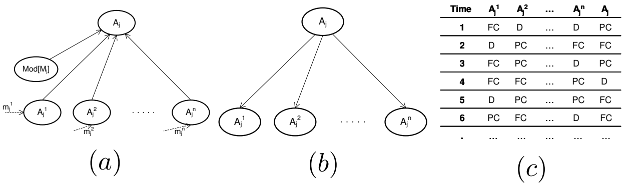
3.1 Relevant Models and Mutual Information

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 $Pr(a_j^1|a_j^*)$ as the probability that a candidate model of j, m_j^n, predicts action a_j^1 when a_j^* is observed in the same time step.

Definition 1 (Relevant Model) *If for a model, m_j^n, there exists an action, a_j^1: $Pr(a_j^1|a_j^*) \geq Pr(a_j^1|\bar{a}_j^*)$, where $a_j^1 \in OPT(m_j^n)$, 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^1, 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. 3(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 to 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. 3(b).



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

Figure 3. (a) The BN in the I-ID of agent i; (b) Equivalent naive BN for classifying outcomes of candidate models to the observation history; (c) Example of the training set used for learning the naive BN for PG. The actions in the last column A_j are observations of i, remaining columns are obtained from candidate 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) [3] between the chance nodes A_j and say, A_j^n, as a measure of the likelihood of the model, m_j^n, in $Mod[M_j]$.

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

$$MI(m_j^n, m_j^*) \quad \overset{def}{=} Pr(A_j^n|A_j)Pr(A_j)log[\frac{Pr(A_j^n|A_j)}{Pr(A_j^n)}] \quad (2)$$

Here, A_j^n is the chance node mapped from the model, m_j^n and A_j are 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 distributions imply the possibility of both observed and predicted actions as well as their relations 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. 3(c). Values of the columns, A_j^1, $A_j^2, \ldots$, 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, $Mod[M_j]$.

3.2 Theoretical Results

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

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

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, m_j^n and m_j^p, there exists an action, a_j^1, such that $Pr_{m_j^n}(a_j^1|a_j^*) \geq Pr_{m_j^p}(a_j^1|a_j^*)$ and $Pr_{m_j^n}(a_j^1|\bar{a}_j^*) \leq Pr_{m_j^p}(a_j^1|\bar{a}_j^*)$, where $a_j^1 \in OPT(m_j^n)$, $OPT(m_j^p)$, the subscript m_j^n or m_j^p denotes the generative model 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 is not shown due to less space).

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 $(m_j^n, m_j^*) \geq MI$ (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.*

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 [10] 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.

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*.

3.3 Algorithm

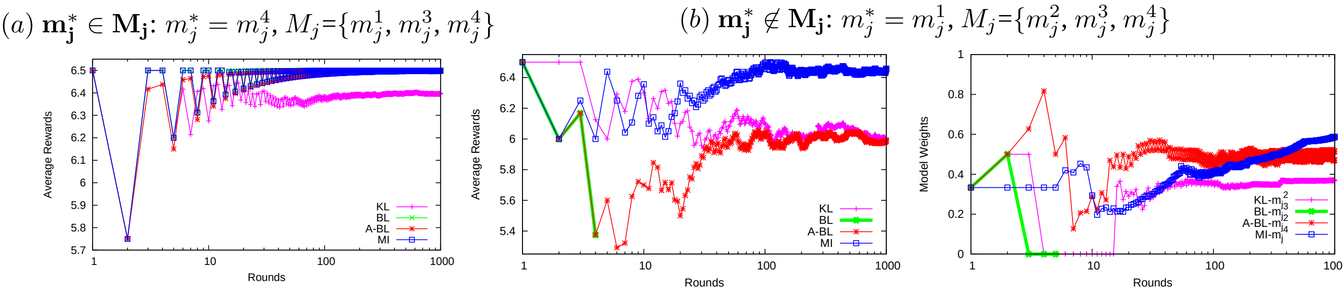
We briefly outline the algorithm for model identification in Fig. 4. In each round t, agent i receives an observation of its opponent j's action (line 1). This observation together with solutions from candidate models of j (line 2), compose one sample in the training set Tr (line 3; see Fig. 3(c)). The 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, $Pr(A_j|A_j^p, m_j^p)$, 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 CPD 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.

```
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 model, $m_{j,t}^p$ $(p = 1, \ldots, n)$ to get actions for the
   chance nodes $A_{j,t}^p$ $(p = 1, \cdots, n)$
3. Add $(A_{j,t}^1, \cdots, A_{j,t}^p, \cdots, A_{j,t}^n, o_i^t)$ 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 CPDs of the chance node $A_j$ using $Pr(A_j|A_j^p, m_j^p)$
9. Normalize $MI(m_j^p, m_j^*)$
10. Populate CPD of the chance node $Mod[M_j]$ using $MI$
```

Figure 4. 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 Performance Evaluation

We evaluate the effectiveness of the algorithm outlined in Fig. 4 in the context of the repeated PG game and re-



(a) $\mathbf{m_j^*} \in \mathbf{M_j}$: $m_j^* = m_j^4$, M_j=$\{m_j^1, m_j^3, m_j^4\}$ (b) $\mathbf{m_j^*} \notin \mathbf{M_j}$: $m_j^* = m_j^1$, M_j=$\{m_j^2, m_j^3, m_j^4\}$

Figure 5. 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.

peated one-shot negotiations as in [11] though simplified. 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), 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 models for the case where $m_j^* \notin M_j$. Due to lack of space, we do not show the changing model weights for the case where $m_j^* \in M_j$.

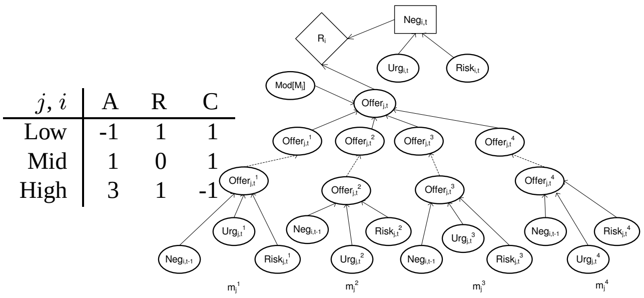
4.1 Repeated Public Good Game

In the PG game, we utilize the I-ID in Fig. 2 to model the interaction. 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. 5(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.

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. 5(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.

4.2 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 successfully or fails, the agents restart a new one on a different item; otherwise, they continue to bargain. Figure 6(a) shows the payoffs of the seller agent when interacting with the buyer. The seller aims to profit 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.

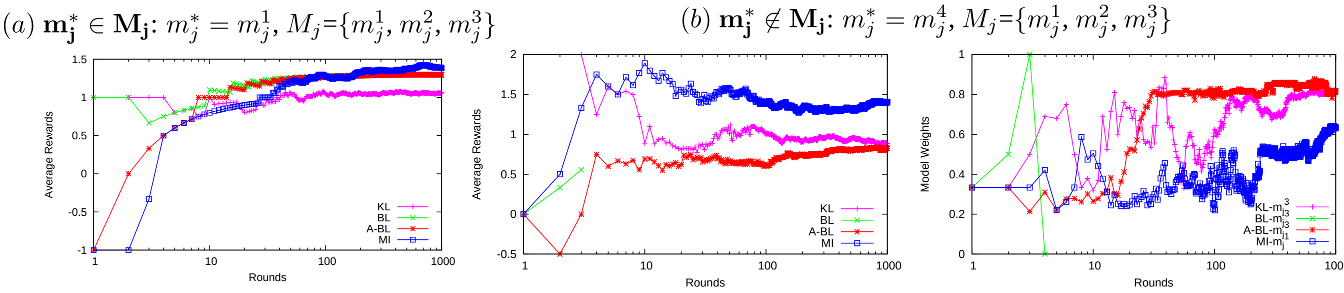


j, i	A	R	C
Low	-1	1	1
Mid	1	0	1
High	3	1	-1

(a) (b)

Figure 6. (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 negotiation with four models ascribed to j.

The idea of using probabilistic graphical models in multiagent negotiation was previously explored in [9]. In a



(a) $\mathbf{m_j^*} \in \mathbf{M_j}$: $m_j^* = m_j^1$, M_j=$\{m_j^1, m_j^2, m_j^3\}$

(b) $\mathbf{m_j^*} \not\in \mathbf{M_j}$: $m_j^* = m_j^4$, M_j=$\{m_j^1, m_j^2, m_j^3\}$

Figure 7. Performance profiles of the MI approach and the changing model weights while repeatedly playing the negotiation game.

similar vein, we model agent i using the I-ID shown in Fig. 6(b). Analogous to [11], 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 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. 7(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 $m_j^* \not\in M_j$, 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 the risk neutral agent m_j^3, which leads to lower average rewards.

5 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, 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 does not affect performance, it merits further investigation.

References

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

[2] D. Carmel and S. Markovich. Learning models of intelligent agents. In *AAAI*, pages 62–67, 1996.

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

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

[5] P. Doshi, Y. Zeng, and Q. Chen. Graphical models for online solutions to interactive pomdps. In *AAMAS*, pages 809–816, 2007.

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

[7] Y. Gal and A. Pfeffer. A language for modeling agent's decision-making processes in games. In *AAMAS*, 2003.

[8] D. Koller and B. Milch. Multi-agent influence diagrams for representing and solving games. In *IJCAI*, pages 1027–1034, 2001.

[9] C. Mudgal and J. Vassileva. An influence diagram model for multi-agent negotiation. In *ICMAS*, pages 451–452, 2000.

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

[11] S. Saha, A. Biswas, and S. Sen. Modeling opponent decision in repeated one-shot negotiations. In *AAMAS*, pages 397–403, 2005.

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

[13] D. Suryadi and P. Gmytrasiewicz. Learning models of other agents using influence diagrams. In *UM*, pages 223–232, 1999.

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