Approximating Value Equivalence in Interactive Dynamic Influence Diagrams Using Behavioral Coverage

Ross Conroy, Yifeng Zeng, Jing Tang
Teesside University, Middlesbrough, UK
{Ross.Conroy, Y.Zeng, x9019186}@tees.ac.uk

Abstract
Interactive dynamic influence diagrams (I-DIDs) provide an explicit way of modeling how a subject agent solves decision making problems in the presence of other agents in a common setting. To optimize its decisions, the subject agent needs to predict the other agents’ behavior, that is generally obtained by solving their candidate models. This becomes extremely difficult since the model space may be rather large, and grows when the other agents act and observe over the time. A recent proposal for solving I-DIDs lies in a concept of value equivalence (VE) that shows potential advances on significantly reducing the model space. In this paper, we establish a principled framework to implement the VE techniques and propose an approximate method to compute VE of candidate models. The development offers ample opportunity of exploiting VE to further improve the scalability of I-DID solutions. We theoretically analyze properties of the approximate techniques and show empirical results in multiple problem domains.

1 Introduction
Extending single-agent dynamic influence diagrams (DIDs) [Howard and Matheson, 1984], interactive DIDs (I-DIDs) [Doshi et al., 2009; Zeng and Doshi, 2012] are a graphical framework for sequential multiagent decision making (planning) under uncertainty. Since I-DIDs deal with decision problems from an individual agents’ perspective, they become a general decision making framework in both competitive and cooperative multiagent settings. Emerging I-DID applications include supplying control policies in automated vehicle routing problems [Luo et al., 2011], developing adversarial models in money laundering activities [Ng et al., 2010], designing intelligent non-player characters in real-time strategy games [Conroy et al., 2015] and so on.

From a subject agent’s viewpoint, I-DIDs need to predict behavior of other agents. The prediction is generally conducted by solving a set of candidate models ascribed to other agents since the true model of the other agents is often unknown particularly in a competitive multiagent setting. The candidate models could be many and are updated when the other agents act and receive new observations over time. Consequently, the subject agent needs to maintain an exponentially growing space of candidate models, which is the main computational complexity towards solving I-DIDs.

Most of the previous I-DID solutions focus on behavioural equivalence (BE) techniques and reduce the model space by grouping candidate models that generate identical policies for other agents [Zeng and Doshi, 2012]. More recently, a fresh idea of model reduction resorts to the concept of value equivalence (VE) and groups models that bring identical expected value to a subject agent [Conroy et al., 2016]. Compared to the BE techniques, the VE method groups models that are either behaviorally equivalent or distinct. Conceptually, VE exhibits large potential to improve the scalability of I-DID solutions. However, identifying VE seems not to be applicable since it requires to compute optimal policies in an I-DID that needs to be built for the subject agent. In an initial investigation, Conroy et al. [2016] learn VE from available data of agents’ interaction by recording their policies as well as values assigned to the policies. In this paper, we compute VE in a general setting where candidate models of other agents are known and an I-DID needs to be built to compute expected values for the subject agent.

Ideally, expected values received by the subject agent are calculated in a complete I-DID that expands all candidate models of other agents. However, this is a bit contradictory since we can’t build the complete model due to computational and memory limits. Hence the challenge is: given the limited model space, how can we compute VE in a sufficiently good manner? To address this, we need to first select a subset of candidate models to build an I-DID for the subject agent, then calculate expected values of optimal policies resulting from the incomplete I-DID.

The model selection is important since it impacts the solution quality of the incomplete I-DID. Intuitively, the resulting I-DID model may generate near optimal policies if the selected models can sufficiently represent the entire set of candidate models. We measure the model representativeness in terms of its solution that prescribes agent’s behavior, and propose a behavioral coverage function to facilitate the model selection process. As the policies resulting from the incomplete I-DID don’t reflect a complete profile of the subject agent, we cannot simply retrieve the expected value from the model. In-
stead, we calculate the expected values for the subject agent in a simulated environment: given the (sub)-optimal policies, the subject agent interacts with other agents who use policies calculated from any of their candidate models. In this context, we make the following contributions:

- We develop a principled framework for VE identification given limited model space. It includes two phases: model selection and value computation.
- We focus on the model selection and propose a behavioral coverage function to choose top-\(K\) models in order to build an incomplete I-DID. We theoretically analyze the proposed function and develop a greedy algorithm for the top-\(K\) model selection.
- We empirically examine VE identification framework, discussing improvements driving a new line of I-DID research.

2 Background: Interactive Dynamic Influence Diagram

We briefly review the I-DID framework with elaboration in the well-studied multiagent tiger problem [Gmytrasiewicz and Doshi, 2005]. We then describe two types of I-DID solutions, namely behavioral equivalence (BE) and value equivalence (VE). For details we refer to [Zeng and Doshi, 2012]

2.1 Representation

I-DIDs are used to represent sequential multiagent decision making problems when a subject agent interacts with other agents under uncertainty and partial observability. I-DIDs model the predicted behavior of other agents by solving a set of their candidate models. Actions of both the agents influence state \(S\) and rewards \(R\). In Fig. 1 we show a level \(l\) I-DID representation for agent \(i\) modeling another agent \(j\) in level \(l - 1\). Level refers to the recursive reasoning of both agents, where level 0 is the lowest by not modeling the actions of others. I-DIDs achieve this by introducing a new node to the framework, the model node, \(M_{j,l-1}\), modeling the decision making process of agent \(j\) at level \(l - 1\). This is accomplished by \(M_{j,l-1}\) containing a candidate set of \(j\)'s models from which their expected behavior can be calculated \(A_j\). The connection between the model node and the predicted behavior is represented by a policy link (the dashed line) connecting \(M_{j,l-1}\) and \(A_j\). Each candidate model contained within \(M_{j,l-1}\) can itself be a level \(l - 1\) I-DID or level 0 DID.

Complexity arises towards modeling with I-DIDs due to the update of the model node built up of all of \(j\)'s candidate models. Such models require updating with every time step via the model update link (the dotted arrow from \(M_{j,l-1}^t\) to \(M_{j,l-1}^{t+1}\) in Fig. 1), as agent \(j\) acts and receives observations over time. The updated models differ in the beliefs that are obtained for a pair of \(j\)'s actions and observations. Agent \(i\) tracks the updates of \(j\)'s models with the number of models growing in a new model node. The number of models grows exponentially for each time step. The number of models in \(M_{j,l-1}^{t+1}\) is up to \(|M_{j,l-1}^t||\Omega_j|||A_j||\Omega_j|\) where \(|M_{j,l-1}^t|\) is the number of models at time step \(t\), and \(|A_j||\Omega_j|\) are the largest spaces of actions and observations respectively.

To solve I-DIDs we replace the model nodes and update links with chance nodes and dependency links. This converts the I-DID into a regular DID allowing for any DID solving technique to be applied. Below we show the multiagent tiger problem to elaborate the I-DID framework.

We show a level 1 I-DID for agent \(i\) considering two candidate models of agent \(j\), \(m_{j,0}^{t,1}\) and \(m_{j,0}^{t,2}\) at Level 0 in Fig 2. The I-DID has been converted to a regular DID where the chance node \(OD\) represents \(j\)'s candidate models differing in beliefs of the tigers location. Solving all candidate models obtains \(j\)'s expected optimal decisions. The conditional probability table (CPT) in Fig. 4 shows agent \(j\)'s optimal decisions of \(GL\) or \(GH\) when the candidate models have been solved for level 0. These optimal decisions can then be mapped into the predicted actions of \(j\) in \(A_j^{t+1}\).

In Fig. 3 we show the model update of \(m_{j,0}^{t,1}\) and \(m_{j,0}^{t,2}\) as agent \(j\) receives one of two possible observations (either \(GL\) or \(GH\)). This requires four new models to be generated in the model node \(M_{j,0}^{t+1}\).

Figure 2: Converted level 1 I-DID for agent \(i\), tiger problem.

Figure 3: Details of the model update link where two models are expanded into four models in \(M_{j,0}^{t+1}\).

Fig. 4 shows he CPT of \(Mod[M_{j,0}^{t+1}]\). For example, the first
row of the CPT shows \( m_{j,0}^{t+1} \) is updated into the model \( m_{j,0}^{t+1,1} \) when agent \( j \) takes the action \( OL \) at time \( t \) and observes \( GL \) at \( t+1 \). As neither \( OR \) nor \( L \) is the optimal decision for \( m_{j,0}^{t+1,1} \), we assign a uniform distribution to indicate \( m_{j,1}^{t} \) does not transform into any of the new models for these actions.

![Figure 4: The CPTs of the chance nodes \( A_j \) and \( \text{Mod}[M_{j,0}^{t+1}] \).](image)

2.2 Solutions
Solving a level \( l \) I-DID includes two steps. We first need to solve \( j \)'s candidate models at level \( l-1 \) and expand them in the I-DID, converting the I-DID into a regular DID (as shown in Fig. 2). After that, we can use any conventional DID technique to solve the converted model. The main complexity is the exponential growth in the number of models over time. Most existing research focuses on two types of model reduction techniques to compress the model space in I-DIDs.

Behavioral Equivalence (BE) groups candidate models with identical policies for agent \( j \) at level \( l-1 \), as defined below.

**Definition 1 (BE)** Two models, \( m_j \) and \( \bar{m}_j \), of agent \( j \), are behaviorally equivalent if \( \text{OPT}(m_j) = \text{OPT}(\bar{m}_j) \), where \( \text{OPT}() \) denotes the solution of the model.

A model solution is generally represented by a policy tree. A depth-\( T \) policy tree contains a set of policy paths, \( T_T = \bigcup h_j^T \), where the policy path, \( h_j^T \), is an action-observation sequence over \( T \) planning horizons. We let \( h_j^T = \{ a_j^0, a_j^{1+1}T-1 \}_{T=0} \), where \( a_j^T \) is null with no observations following the final action.

Value Equivalence (VE) groups models that generate the same expected value for agent \( i \) at level \( l \), as defined below.

**Definition 2 (VE)** Two models of agent \( j \), \( m_{j,l-1} \) and \( \bar{m}_{j,l-1} \), are value equivalent if \( V^T(m_{i,l}|m_{j,l-1}) = V^T(\bar{m}_{i,l}|\bar{m}_{j,l-1}) \), where \( V^T() \) is the expected values for agent \( i \) given that the model \( m_{i,l} \) is expanded by replacing \( m_{j,l-1} \) with \( \bar{m}_{j,l-1} \) (or vice versa) in a set of \( j \)’s candidate models.

Expected value of agent \( i \)'s model \( m_{i,l} \) computed in Eq. 1.

\[
V^T(m_{i,l}) = \rho(b_{i,l}, a^*_i) + \sum a_o Pr(a_o|b_{i,l}, a^*_i)V^T-1(m_{i,l})
\]

where \( \rho(b_{i,l}, a^*_i) = \sum b_{i,l}(s, m_{j,l-1}) \sum R_i(s, a^*_i, a_j) \times Pr(a_j|m_{j,l-1}) \).

Here, \( b_{i,l}(s, m_{j,l-1}) \) is the agent \( i \)'s belief over the physical states and possible models of \( j \) at level \( l-1 \), \( a^*_i \) is \( i \)'s optimal action and \( m_{j,l} \) is the updated model of agent \( i \) containing the updated belief at the next time step.

Note that agent \( i \) assigns some probability mass to every candidate model of agent \( j \) in the model node \( \text{Mod}[M_{j}] \).

When the replacement \((m_{j,l-1} \leftrightarrow \bar{m}_{j,l-1}) \) occurs, we transfer the probability mass over the two models. According to Eq. 1, the VE models could be either behaviorally equivalent or distinct. Hence using VE, compared to BE, could lead to more reduction in the model space of agent \( j \). The most recent I-DID research [Conroy et al., 2016] proposes the VE concept, but doesn’t develop applicable I-DID solutions.

3 Model Selection and Value Computation
VE determination needs to compute the expected value of agent \( i \), which requires building an I-DID for \( i \) and solve the model accordingly. An exact solution expands all candidate models of agent \( j \) to build a complete I-DID. However, this is not applicable. In this paper, we approximate the VE identification given the limited model space in an incomplete I-DID.

3.1 Value Equivalence Identification Framework

Let \( M_{j,l-1} \) be the set of all candidate models of agent \( j \). Through VE, we aim to reduce the set into the limited space containing \( K \) models \( M^K_{j,l-1} \). Alg. 1 shows a principled framework for identifying VE of candidate models, which results in the compressed model space.

**Algorithm 1 VE Identification Framework**

```plaintext
1: function FRAMEWORK(m_{i,l}, M_{j,l-1}, K, N)
2: Initialize M^K_{j,l-1} = ∅, Ite=0
3: while |M^K_{j,l-1}| < K ∧ (Ite + |M^K_{j,l-1}| < |M_{j,l-1}|) do
4: New M^K_{j,l-1} ← SelectModel(M_{j,l-1}, K)
5: for all m_{j,l-1}, \bar{m}_{j,l-1} ∈ M^K_{j,l-1} do
6: Build \bar{m}_{j,l-1} given \bar{m}_{j,l-1} ← m_{j,l-1}
7: Build m_{j,l-1} given m_{j,l-1} ← m_{j,l-1}
8: \bar{T}_i ← m_{i,l}.Solve, T_i ← m_{i,l}.Solve
9: V(\bar{m}_{i,l}|m_{j,l-1}) ← ComputeValue(\bar{T}_i, M^K_{j,l-1}, N)
10: V(m_{i,l}|m_{j,l-1}) ← ComputeValue(T_i, M^K_{j,l-1}, N)
11: if V(\bar{m}_{i,l}|m_{j,l-1}) = V(m_{i,l}|m_{j,l-1}) then
12: M^K_{j,l-1} ← \{m_{j,l-1}\}
13: Ite = Ite + 1
14: return M^K_{j,l-1}
```

In this framework, we first develop a model selection function to choose a subset of \( j \)'s candidate models based on which we can build incomplete I-DIDs for agent \( i \) (line 4). We then retrieve optimal policies of agent \( i \) by solving the built I-DIDs when a mutual replacement is conducted in the subset (lines 6-8). We can use the previous I-DID techniques to solve the models (line 8). Subsequently, we compute the expected values of agent \( i \)'s policies through a value computation function that evaluates the policies over \( N \) simulations (lines 9-10). The value comparison may prune the VE
models and further compress the subset (lines 11-12). We repeat the process until the limited model space is filled with K models or all the candidate models have been searched.

3.2 Top-K Model Selection

Without a complete I-DID built by expanding all candidate models, we can’t guarantee an optimal policy will be used to compute the expected values of agent i. Given the limited model space, we aim to provide reasonably good policies by selecting a proper subset of j’s candidate models to build I-DIDs. We proceed to develop such a selection mechanism.

Intuitively, we need to choose K models whose joint solutions (representing agent j’s behavior) have the largest coverage of solutions of an entire set of j’s candidate models. In other words, the quality of i’s policies resulting from the I-DID may not be significantly compromised if agent i considers as many as possible representative behaviors of j. The representative behavior occurs frequently in agents’ interactions. Inspired by this, we introduce a behavioral coverage function that measures the similarity between models.

The similarity between models mj and mj, denoted by \( w(m_j, m_j') \), is defined by how similar the policy generated by mj is to that of mj'. It is calculated in Eq. 2.

\[
  w(m_j, m_j') = \sum_{h_j^T \in T_j} \sum_{h_j^T' \in T_j'} \text{sim}(h_j^T, h_j')
\]

where \( \text{sim}(h_j^T, h_j') \) counts the number of identical actions given same observations at each time step.

Then, \( \sum_{m_j' \in M^K_{j, l-1}} w(m_j, m_j') \) measures how much of the model mj is covered by the selected K models. We aim to find a set of top-K models, \( M^K_{j, l-1} \), that have the largest behavioral coverage of the entire model space \( M_{j, l-1} \). Formally the top-K model selection is formulated as one optimization problem below.

Given: \( M_{j, l-1}, K \)

Objective: 

\[
  \max_{M^K_{j, l-1} \subseteq M_{j, l-1}, |M^K_{j, l-1}| = K} \sigma(M^K_{j, l-1}) = \sum_{m_j \in M_{j, l-1}} \sum_{m_j' \in M^K_{j, l-1}} w(m_j, m_j')
\]

We observe that the model selection is a complex combinatorial optimization problem with a single objective. We prove it to be NP-hard.

**Proposition 1** The top-K model selection problem formulated in Eq. 3 is NP-hard.

**Proof.** We prove it by converting Eq. 3 into a unit cost version of the budgeted maximum coverage problem (UBMC) [Khuller et al., 1999]. Given a unit cost version of the UBMC problem instance \( \varphi \) : a collection of sets \( S = \{S_1, S_2, \ldots, S_m\} \) with a unit cost \( C \), a domain of elements \( X = \{x_1, x_2, \ldots, x_n\} \) with associated weights \( \{z_1, z_2, \ldots, z_n\} \), and a budget \( B \), we can construct a top-K model selection instance \( \omega \) by setting \( K = [B/C] \) and \( \sigma(S') \) corresponds to the total weight of the elements covered by \( S' \). Since \( S' \) is the set having a maximum weight in \( \varphi \) iff \( S' \) is the top-K model set of \( \omega \). As the UBMC problem has been proved to be NP-hard, the top-K model selection problem is NP-hard as well.

It is rather hard to solve the model selection problem. Meanwhile, we notice that the selection function \( \sigma(M^K_{j, l-1}) \) is a monotone submodular function [Nemhauser et al., 1978]. Let \( \mathcal{V} \) be a finite set. A set of function \( F: \mathcal{V} \rightarrow \mathbb{R} \) is called submodular if it satisfies the diminishing returns property, \( F(B \cup s) - F(B) \geq F(B \cup s) - F(s) \), for all \( B \subseteq B \subseteq \mathcal{V} \) and \( s \notin \mathcal{V} \). \( F(B \cup s) - F(B) \) is the marginal increase of \( F \) when an element \( s \) is added into \( B \). Submodularity characterizes the property that unifying elements to a small set \( B \) provides more than doing it to a larger set \( \tilde{B} \).

Naturally, \( \sigma(M^K_{j, l-1}) \) is monotone as the model coverage increases with a larger set of candidate models. It is also submodular. Intuitively, the increment when adding a new model into a small set of top-K1 models will be larger than the increment when adding it to a large set of top-K2 models, where \( K_1 < K_2 \), since the behavior exhibited by the new model might have already covered by those models that are in the larger set but not in the small set. This is the diminishing returns property. We present the property of \( \sigma(M^K_{j, l-1}) \) in Proposition 2.

**Proposition 2** The model selection function \( \sigma(M^K_{j, l-1}) \) is monotone and submodular.

The monotone submodular property suggests a greedy algorithm with theoretical guarantees for optimizing the model selection function [Nemhauser et al., 1978]. In Alg. 2, the greedy algorithm starts with an empty set and computes behavioral coverage of every model (lines 2-4). Then repeatedly adds the model incurring the largest marginal coverage increasing the model set until \( |M^K_{j, l-1}| = K \) (lines 5-7). The algorithm achieves near-optimal solutions of top-K models with a \((1-\frac{1}{e})\) approximation on optimal behavioral coverage.

Since the greedy algorithm needs to check all of the candidate models in every round (line 6), the time complexity is \( O(K|M_{j, l-1}|B(\sigma(\cdot))) \), where \( B(\sigma(\cdot)) \) the run time for computing the model coverage.

**Algorithm 2** Model Selection

1: function SELECTMODEL(M_{j, l-1}, K)
2: \( M^K_{j, l-1} = \emptyset \)
3: for all \( m_j \in M_{j, l-1} \) do
4: \( \quad \) Compute \( \sigma(m_j) \)
5: for \( l = 1 \) to \( K \) do
6: \( \quad m_j \leftarrow \arg \max_{m_j} [\sigma(M_{j, l-1} \cup m_j) - \sigma(M_{j, l-1})] \)
7: \( \quad M^K_{j, l-1} \leftarrow M^K_{j, l-1} \cup m_j \)
8: return \( M^K_{j, l-1} \)

**Remarks.** We note that many previous BE techniques [Zeng and Doshi, 2012] have the same purpose of selecting a subset of candidate models to develop I-DIDs of high quality. However, the techniques prune the models through a pair check of behavioral equivalence and don’t consider impact of individual models on the global behavioral coverage. As demonstrated in empirical study (in
Section 4.1, top-K model selection technique provides better I-DID solutions.
Selection of K value often depends on model space allowed in I-DIDs so that the I-DIDs can be solved for a specific planning horizon. A trade-off between quality and scalability.

3.3 Value Computation
Given top-K models, we build an I-DID based on which we proceed to identify VE of models in the selected set. As the incomplete I-DID is built from a subset of j’s candidate models, the resulting policy for agent i may not be the same as that computed from a complete I-DID expanded by all j’s models. The expected value of the incomplete I-DID is not a good measurement of agent i’s policy since unexpected behavior of agent j that may occur in their interactions may not be considered by agent i. Thus, we compute the expected value for agent i by simulating how i interacts with agent j. The value is counted as the average rewards that agent i receives when it interacts with agent j over a number of times. This is well matched with how I-DID solutions are evaluated in the previous I-DID research. To have the self-contained paper, we present a value computation in Alg. 3.

Given agent i’s beliefs over j’s models, we sample a model of j and solve the model to obtain its policies (lines 4-5). Then, agents follow their policies in the interactions (lines 6-7). We conduct N simulations and compute the expected value of i’s policies as the average reward of the simulations.

4 Experimental Results
We implemented the framework (in Alg. 1) to prune VE models of agent j. The implementation replaces the previous BE pruning methods in solving agent i’s I-DID [Zeng and Doshi, 2012]. We first verify the performance of the top-K model selection in Alg. 2, which itself can be used to solve I-DID, in comparison to the state-of-art BE methods. We then evaluate the entire VE identification framework in two large problem domains. One is the UAV benchmark (|S|=81, |A|=5 and |Ω|=5) [Zeng and Doshi, 2012] - currently the largest problem domain studied in I-POMDP/I-DID based multiagent planning research while the other is a real-world game domain of StarCraft 1 (|S|=16, |A|=3 and |Ω|=4). We build level 1 I-DIDs for agent i. To compare different I-DID techniques, we compute the average rewards of agent i when it plays against agent j by executing their policies solved from the I-DIDs.

1http://eu.blizzard.com/en-gb/games/sc/
(b) Model Selection Time

Figure 6: Performance of VE-based I-DID techniques in UAV

\( T = 5 \) and \( |M_{j,t-1}|=25 \) \hspace{1cm} \( T = 7 \) and \( |M_{j,t-1}|=20 \)

Figure 7: Performance of VE techniques in StarCraft

\( T=5 \) and \( |M_{j,t-1}|=10 \)

Figure 6: Performance of VE-based I-DID techniques in UAV

(a) \( T = 5 \) and \( |M_{j,t-1}|=25 \) \hspace{1cm} \( T = 7 \) and \( |M_{j,t-1}|=20 \)

Figure 7: Performance of VE techniques in StarCraft

(a) \( T=5 \) and \( |M_{j,t-1}|=10 \) \hspace{1cm} \( T=5 \) and \( |M_{j,t-1}|=10 \)

(b) Model Selection Time

main has partial observability by hiding the true state of the game to the player, requiring the player to make observations influenced by the true state. Human vs Human games are incredibly complex with players required to maintain focus in many problem areas such as resource management and battle tactics. In this paper, we focus on a typical 3 vs 3 unit battle.

As a small number of \( j \)'s candidate models are tested, we implemented an optimal search algorithm (OS) to select top-

K models. OS compares behavioral coverage values for every combination of \( K \) models. Fig. 7a shows that VE+GS performs closely with VE+OS for solving I-DIDs. It has slightly better performance than VE+OS in some cases. This is because the approximation of VE+GS may introduce models that concentrate on a very small number of representative behaviors. However, OS distributes the focus (\( i \)'s beliefs) over multiple types of behaviors to optimize the coverage. Particularly in game-play, players often focus on a specific gaming pattern, which is confirmed in the available game replay data. Agent \( i \) will be highly rewarded by successfully predicting this single type of behavioral pattern.

Fig. 7b compares model selection time. RS is much faster than others, but results in the lowest rewards. OS starts with similar durations; however, as \( K \) increases it increases much faster than GS. We don’t show the time of computing values in Alg. 3 since it is similar among all the three methods due to similarly sized I-DIDs generated. Although DMU model selection is compared in model selection, the resulting I-DID's model space is too large to be solved. Thus showing the improved scalability of our model reduction method.

5 Related Works

I-DIDs are a graphical representation of finitely-nested interactive partially observable Markov decision processes (I-POMDPs) [Gmytrasiewicz and Doshi, 2005]. Exponential growth in candidate models of other agents adds significant complexity to solving I-DIDs.

Currently the accepted methods exploit BE to reduce the model space within I-DID solutions [Zeng and Doshi, 2012]. Multiple works examine policy trees to develop methods to reduce the model space. Doshi et al. [Doshi et al., 2010] introduces an \( \epsilon \)-subjective equivalence method. \( \epsilon \)-subjective equivalence seeks to prune the model space of I-DIDs by pruning equivalent models comparing agents future action observation paths. Zeng et al. [Zeng et al., 2011] cluster models by comparing only a partial set of paths within the policy trees of other agents \( j \). Online I-DID solutions of expand the true behavior of other agents from interactions pioneered by Chen et al. [Chen et al., 2015]. Learning behavior of agents from available data [Conroy et al., 2015] provides knowledge towards refining the model space in I-DIDs. BE techniques have aided in the usefulness of I-DIDs and shown potential data-driven learning towards real-world applications [Luo et al., 2011; Conroy et al., 2015]. In line with the recent work of Albrecht et al. [Albrecht et al., 2015] with type based methods our method of model reduction can be generalised as an example of such a method.

Most relevant work of utility equivalence techniques in a social simulation setting of class bullying [Pynadath and Marsella, 2007], shows a minimal number of mental models that could be maintained through grouping utility equivalent models. In the same vein, Conroy et al. 2016 show benefit of VE-based I-DID solutions where VE models can be found from available interaction data without building I-DIDs.

Graphical models of cooperative decision making scenarios utilizing frameworks such as decentralized POMDPs [Seeken and Zilberstein, 2008] remain relatively unexplored while factored representations of the state space are becoming prevalent [Oliehoek et al., 2008]. Such factored representations allow for solutions to decentralized POMDPs amongst multiple agents by exploiting the structure of interactions amongst such agents [Oliehoek et al., 2013]. Factored representations in dynamic Bayesian networks to project agents’ beliefs forward, then expectation-maximization learning of stochastic finite state machines was utilized by Pajarinen and Peltonen [2011].

6 Conclusion and Future Work

Value equivalence exhibits scalability improvement over the BE techniques for solving I-DIDs. More importantly, VE methods are developed in consideration of expected values of a subject agent so that they can directly measure the solution quality. Given the limited model space, the VE techniques can’t avoid approximation in the VE identification. However, as demonstrated in the empirical study, the top-\( K \) model selection provides sufficiently good I-DID solutions.

Immediate VE-based I-DID research may consider improving the effectiveness of the top-\( K \) model selection or efficiency of value computation. The behavioral coverage function can be extended to include other factors. For example, as suggested in the above tests in Starcraft, it is worth focusing on highly rewarded behavioral patterns. However, holding a monotone submodular function is important since it introduces efficient approximation with theoretical guarantee on solution quality. Another interesting direction is the improvement of computing expected values of agents’ policies in multiagent settings by reducing the number of samples.
References


