Loss Bounds for Approximate Influence-Based Abstraction

Elena Congeduti
Computer Science
Delft University of Technology
E.Congeduti@tudelft.nl

Alexander Mey
Computer Science
Delft University of Technology
A.Mey@tudelft.nl

Frans A. Oliehoek
Computer Science
Delft University of Technology
F.A.Oliehoek@tudelft.nl

ABSTRACT

Sequential decision making techniques hold great promise to improve the performance of many real-world systems, but computational complexity hampers their principled application. Influence-based abstraction aims to gain leverage by modeling local subproblems together with the 'influence' that the rest of the system exerts on them. While computing exact representations of such influence might be intractable, learning approximate representations offers a promising approach to enable scalable solutions. This paper investigates the performance of such approaches from a theoretical perspective. The primary contribution is the derivation of sufficient conditions on approximate influence representations that can guarantee solutions with small value loss. In particular we show that neural networks trained with cross entropy are well suited to learn approximate influence representations. Moreover, we provide a sample based formulation of the bounds, which reduces the gap to applications. Finally, driven by our theoretical insights, we propose approximation error estimators, which empirically reveal to correlate well with the value loss.

KEYWORDS

Sequential decision making; Multiagent systems; Best-response problems; State abstraction; Learning abstractions; Loss bounds

1 INTRODUCTION

Sequential decision making methods have the potential to improve control in distributed or networked systems that can involve many agents and complex environments. However, applying these methods in a principled manner is often difficult due to computational complexity. One idea to improve scalability is using abstractions, compressed representations of the sufficient information for an agent to perform optimal decisions. Many abstraction approaches have been suggested [3, 17, 23] but in order to ensure small value loss, they only allow abstracting states of the environment with similar dynamics. A body of work on localized abstractions [4, 5, 38, 42] tries to overcome this issue, allowing to abstract large part of the system away. These approaches leverage the sparse interaction structures of many real-world domains where the agent's reward and observations depend directly only on few (local) state variables.

Influence-based abstraction (IBA) [33] generalizes over these approaches and provides a unified framework for localized abstractions for a general class of multiagent problems. The basic idea is to decompose structured multiagent systems into small submodels for each agent where only few local variables are included. The 'influence' summarizes the indirect effects of other agents policies and the rest of the environment on the local variables of a single agent. Unlike existing methods to compute best responses [26, 30, 34] which involve reasoning over the entire state space and other agents action-observation histories, IBA ensures that the decision maker needs to reason only over few local state variables. As a result, the best-response problem can be solved much more efficiently in the local model without any loss in value. This approach may not only enable speedups in best-response problems but may serve also as the foundation for scalable multiagent approaches based, for instance, on searching in the space of influences [42].

However, even though IBA provides a lossless abstraction, in general, deducing an *exact representation* of the influence itself is computationally intractable even for simple problems. Given the rapid progress in learning methods for sequence prediction [2, 20, 22], we envision that inducing *approximate representations* of the influence offers a promising approach to scalable solutions. This raises a fundamental question: to what extent are such existing methods (and their proxy-loss functions) aligned with the actual influence-base abstraction objective of achieving near-optimal value?

In this work we address this issue, showing that there is theoretical and empirical evidence that employing existing sequence predictors can lead to good approximate influence representations. In particular, we derive a performance loss bound as well as a probabilistic version that provides quality guarantees on approximate influence to ensure near-optimal solutions. Like bounds based on the Bellman error, our bounds themselves are computationally intractable for large problems. This is to be expected: the performance loss is characterized by the quality of each possible prediction. Therefore, deriving an exact bound essentially corresponds to computing the exact generalization error of the employed machine learning model. Nevertheless, we show that a neural network trained with cross-entropy loss seems well suited for the prediction task, as the training objective is aligned with the bound we derived. Finally, we empirically demonstrate that it is possible to compute statistics, based on the empirical test error of approximate influence, that correlate well with the actual value loss.

In summary, our contributions in this paper are:

(1) Sufficient conditions for the quality of the influence approximation in terms of value loss.
(2) Discussing how these conditions are aligned with usual optimization of neural networks trained with cross entropy loss.
(3) An empirical evaluation showing that optimizing cross entropy loss of influence predictors leads to lower performance loss.

2 RELATED WORK

Abstraction has been widely studied as a technique to accelerate learning and improve scalability in complex (multiagent) problems. Many abstraction approaches have been suggested [1, 14, 15, 17, 23, 35] but they all share a common limitation: in order to guarantee bounded loss, the ground states in the same abstract states (which are clusters of ground states) must all share similar transition probabilities. However, when we want to abstract away entire sets of state variables such guarantees will typically not be possible. Influence-based abstraction differs essentially from these kind of abstractions. In fact, it enables abstracting entire factors still providing guarantees of optimality when the aggregated states have typically very different transition probabilities.

3 BACKGROUND

First we outline the background on sequential decision making.

3.1 Sequential Decision Making Framework

We consider a general class of multiagent problems that can be modeled as partially observable stochastic games [19].

Definition 1 (POSG). A *Partially Observable Stochastic Game* is a tuple $M = (n, S, A, O, T, \Omega, R, h, b^0)$ where n agents interact in the finite state space S. $A = A_1 \times \cdots \times A_n$ and $O = O_1 \times \cdots \times O_n$ are the finite spaces of joint actions and observations. T models the transition probabilities as $T(s'|s, a) = \mathbb{P}(s'|s, a)$, namely the probability of resulting in state s' when actions $a = (a_1, \ldots, a_n)$ are chosen in state s. Ω is the observation distribution, $\Omega(o'|a, s') = \mathbb{P}(o'|a, s')$, the probability of receiving observations $o = (o_1, \ldots, o_n)$ when performing actions a results in state s'. $R(s, a) = (R_1(s, a), \ldots, R_n(s, a))$ specifies the immediate reward function for each agent for taking actions a in state s. Finally, h defines the process horizon and b^0 the initial state distribution.

In a POSG, each agent i employs a policy π_i that corresponds to a possibly stochastic map from the action-observation history to the action space. A joint policy $\pi = (\pi_1, \ldots, \pi_n)$ is a tuple of policies for each of the agents. Given a joint policy π, we can define the corresponding expected cumulative reward function for an agent i: $V_i^\pi = \mathbb{E}\left[\sum_{t=1}^h R_i^t \mid \pi, b^0\right]$.

Different optimization problems and corresponding solution concepts can be considered according to the adversarial or cooperative nature of the agents. In the first case, the problem consists of searching for a Nash equilibrium (NE)[27]: a joint policy $\pi = (\pi_i, \pi_{-i})$ such that each agent i draws no advantage from deviating from its policy π_i given the policies of the others π_{-i}. In cooperative settings modeled by decentralized partially observable Markov decision processes (Dec-POMDPs) [7, 29], the solution consists of the optimal join policy π maximizing the value function shared by all players. Many solutions methods for searching NE (e.g. fictitious play [6, 11], double oracle [25], parallel Nash memory [31]) and for optimal policies for Dec-POMDPs (e.g. influence search [42]) use best-response computations as their inner loop. A best response policy π_i^* for agent i against fixed policies π_{-i} for the other agents maximizes the value function $\max_{\pi_i} V_i^{(\pi_i, \pi_{-i})}$.

However, computing best responses implies reasoning over the entire state space and other agents beliefs [26]. To ease this task we want to exploit the structural properties of the environment to build local abstractions. Frequently, in fact, the state space can be thought as composed of different state variables, or *factors* [10]. In this case, the model is called *factored POSG* and every state $s \in S$ can be represented as a tuple of factors instantiations $s = (x_1, \ldots, x_m)$. This structure allows to decompose the system into (weakly) coupled subproblems, the *local models*, for single agents including only few factors. Thus, the idea is to define an equivalent problem to the best response in which the agent needs only to reason over the factors included in the local model.

3.2 Influence-Based Abstraction

Influence-based abstraction [33] formalizes the concept of the local model for an agent in a factored POSG and provides the formal framework of the local abstraction for the best-response problem.

To simplify the discussion, we restrict to locally fully-observable POSGs [18]. We use capital letters to denote state variables and small letters for variable instantiations.

Definition 2 (Fully Observable Local Model). Given a factored POSG, a *Fully Observable Local Model* for an agent is a subset of fully observable state variables, called *modeled factors*, including all the state variables directly affecting the observations and reward. We denote the *local state*, that is the collection of all the modeled factors, as X and the complementary collection of *non-modeled factors* as Y.

The idea of the local model is to exclude those factors that are not necessary to compute the best response from the subset of variables over which the agent needs to reason and that can therefore be abstracted away. In a local model, we distinguish between modeled factors that are only affected by other factors and actions that are modeled, called *only-locally affected factors*, and the *non-locally affected factors* that are modeled factors affected by at least one factor or action of the external part. We denote by X_{loc} the collection of the only-locally affected factors.

Definition 3. We refer to the non-modeled factors or external actions that directly exert an influence on at least one of the modeled factor as the *influence sources* Y_{src}. We define the *influence destinations* X_{dest} as the modeled factors that directly experience the influence of the external part of the system through the influence sources.

Thus, the local state X can be thought as $X = (X_{\text{loc}}, X_{\text{dest}})$.

In other words, the local model comprises only few modeled-factors. Some of them are influenced by the external part and therefore are called influence destinations. The non-modeled factors or actions that directly exert this influence on the modeled factors are accordingly the sources of the influence. See Section 3.2 in [33] for the formal definitions of the general, not fully-observable, case.

When abstracting away the non-modeled factors Y, the dynamics of X_{dest} become non-Markovian. Accordingly, the local state transitions are not well defined. To define the local state dynamics, we need to include as part of the abstract state space the history of relevant modeled actions and factors to infer the influence sources Y_{src}. The notion of *d-separating set* (d-set) [9] formalizes this concept. The d-set at time t, D^t contains the histories of the modeled factors and actions necessary to predict the influence sources Y_{src}^t.

Definition 4 (Influence Point). The *exact influence point* (EIP) $I = (I^0, \ldots, I^{h-1})$ is a collection of conditional probability distributions $I^t(Y_{src}^t \mid D^t) \triangleq \mathbb{P}(Y_{src}^t \mid D^t, \pi_{-i}, b^0)$ of the influence sources Y_{src}^t given the possible instantiations of the d-set.

We refer to Section 4.1 in [33] for extensive definitions of influence point and d-separating set. The local model dynamics is formalized by the *influence-augmented local model*, a factored Markov decision process (MDP) where the state space consists of the local state augmented with the d-set.

Definition 5 (IALM). Consider a fully observable local model for agent i in a factored POSG $M = (n, S, A, O, T, \Omega, R, h, b^0)$, that identifies modeled factors X. Fix the policies π_{-i} for the other agents. An *Influence Augmented Local Model* $\mathcal{M}_i = (\bar{S}_i, A_i, \mathcal{T}_i, R_i, h, b^0)$ is a factored MDP where the action space A_i and rewards R_i correspond to the POSG agent i's actions and rewards. The augmented state space $\bar{S}_i$ consists of local states and d-separating sets as $\bar{s}^t = (x^t, d^t) = ((x_{loc}^t, x_{dest}^t), d^t)$. The transition functions are derived as

$$\mathcal{T}_i(\bar{s}^{t+1} \mid \bar{s}^t, a^t) = T(x_{loc}^{t+1} \mid x^t, a^t) \sum_{y_{src}^t} T(x_{dest}^{t+1} \mid x^t, y_{src}^t, a^t) \, I^t(y_{src}^t \mid d^t) \tag{1}$$

This model defines an equivalent problem to the best-response problem for agent i against π_{-i}. Namely, the optimal policy for the MDP defined by the IALM $\mathcal{M}_i$ corresponds to the best response against policies π_{-i} in the factored POSG M. The claim and complete proof that IBA provides a lossless abstraction can be found in Section 6 [33].

3.3 Influence in Planetary Exploration

To give a concrete intuition of influence-based abstraction we use a version of the planetary exploration environment [42]. A rover has to explore a planet, and its navigation may be guided by a plan from a satellite. The goal of the rover is to move until it reaches a target site, where it will collect a positive reward. However, any time it fails to step forward, it will receive a penalty. The satellite might help the rover by providing a plan which increases the likelihood of a successful step.

Figure 1 shows a dynamic Bayesian network (DBN) [10] that compactly represents the problem. At each time t the rover can observe its position *pos* and if a plan *pl* was available at the previous time step. The rover's local state consists of $X = (pos, pl)$. The plan *pl* is the influence destination since it is directly affected by the non-modeled satellite action a_{sat} which is accordingly the influence source. Contrarily, the position *pos* is only directly affected by local variables. The decisions of the satellite might depend on its level of battery, the *charge*, which only indirectly affects the local model through the satellite actions. Potentially, the decision-making problem of the satellite might depend on a conceivably larger number of variables: it has to manage its own resources, send plans to other rovers etc. However, the only information the rover needs to retrieve to act optimally is whether the satellite will make available a routing plan at the next step. Therefore, it can abstract away all the other state variables and try to *infer* the satellite's action, given all the relevant information that it has in the local model. In this scenario, it turns out that all it has to remember is the history of

the availability of plans at each time step. Therefore the influence point consists of the distribution of the satellite's actions $Y_{src}^t = a_{sat}^t$, given the local history of the plan $D^t = (pl^0, \ldots, pl^t)$.
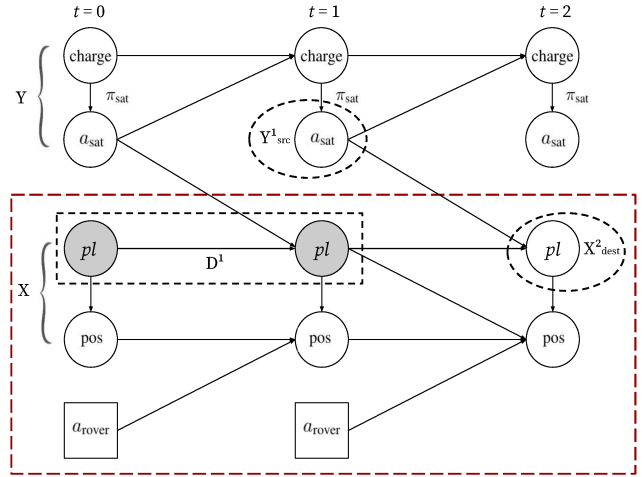

Figure 1: A DBN of the planetary exploration domain. The dotted red square delimits the local model of the rover, including the modeled factors X. The external part comprises the non-modeled factors Y. Among them, the satellite action a_{sat} directly affects the availability of a plan pl and therefore is the influence source. The gray circles constitute the d-set D^1 to infer the action a_{sat} at time $t = 1$. Namely, the history of the plan (pl^0, pl^1) retains the sufficient local information to infer the satellite action a_{sat}^1. The influence on the local model at the next time step $t = 2$ is experienced directly by the modeled factor plan pl^2 that is thus the influence destination.

4 PLANNING WITH APPROXIMATE INFLUENCE REPRESENTATIONS

IBA has the potential to enable faster planning in complex domains where using the entire model would typically be too heavy and computational demanding with no information loss. However, exact influence computation requires solving a large number of intractable inference problems. In fact, in most applications, the size of the d-set increases linearly in time, resulting in an exponentially increasing number of instantiations of the d-set. This means that representing an EIP takes exponential space, and computing an EIP requires solving exponentially many and possibly hard inference problems. This motivates the idea of using *approximate influence point* (AIP) representations: we instead use advances in machine learning that allow us to generalize AIPs over d-separating sets. The overall idea consists of transforming the global model into a new approximate influence-augmented local model at which any solution method can be applied. Our method is therefore complementary to the planning method. The advantage lies in the reduction of the problem size and therefore in computational complexity.

The approach is straightforward but sketched in Algorithm 1 for completeness: first we collect samples of the d-set D^t, the local

history sufficient to predict the influence sources Y_{src}^t in the global model using an exploratory policy π_i^{Exp} for agent i (lines 1-3). In the second step, we learn the approximate influence point $\hat{I}^t$ using the N trajectories of (D^t, Y_{src}^t) as training set (lines 4-5). Then, we construct the approximate local transitions $\hat{\mathcal{T}}_i$ using the learned influence point $\hat{I}$ according to 1 (line 6). Then, we assume that the resulting local problem might be solved by simulation-based planning or RL approaches to derive a near optimal best-response $\hat{\pi}_i^*$ for agent i (line 7-9). Note that in line 8 any solution method can be used to solve the MDP problem. The idea is that this phase uses $M \gg N$ episodes, thus making the initial cost of using the expensive global simulations negligible.

While planning with a learned approximate influence could lead to significant speedups, there is also a risk as the approximate influence point may induce inaccurate state transitions, leading to a loss in value. A key question therefore is: what constitutes good AIPs? What conditions need to hold on $\hat{I}$, for $\hat{\pi}_i^*$ to be close to optimal?

Algorithm 1 Exact Best Response to Approximate Influence

Require: exploratory policy π_i^{Exp}, policies π_{-i}
  1: {Simulation:}
  2: Create a dataset of N trajectories of the global model
  3: Extract $\left(d_k^t, y_{src,k}^t\right)_{k=1:N, t=1:h}$.
  4: {Train a Prediction Algorithm:}
  5: Learn approximate influence $\hat{I}^t$ from $\left(d_k^t, y_{src,k}^t\right)_{k=1:N}$ for any
     $t = 0, \ldots, h$
  6: Compute approximate local transitions $\hat{\mathcal{T}}_i$ from $\hat{I}$
  7: {Solve the local planning/RL problem:}
  8: Compute $\hat{\pi}_i^* = \text{MDPSolver}(\hat{\mathcal{T}}_i)$
  9: **return** $\hat{\pi}_i^*$

---

5 THEORETICAL LOSS BOUNDS

Here we want to discuss guarantees on the value loss when using AIPs instead of EIPs to derive best-response policies. Such kind of results allow us to derive conditions for approximate influence to yield near-optimal solutions.

Formally, we consider two IALMs $\mathcal{M} = (\bar{\mathcal{S}}, A, \mathcal{T}, R, h, b^0)$ and $\hat{\mathcal{M}} = (\bar{\mathcal{S}}, A, \hat{\mathcal{T}}, R, h, b^0)$ sharing the same augmented state space, action space and rewards. They differ only in the transition functions $\mathcal{T}$ and $\hat{\mathcal{T}}$ that are induced respectively by the EIP I and an AIP $\hat{I}$ according to 1. We see $\hat{\mathcal{M}}$ as an approximation to $\mathcal{M}$. We omit the subscript i when referring to an IALM for agent i to ease the notation.

5.1 L^1 Loss Bound

We introduce first the necessary notation. We use $\mathcal{V}^*$ to denote the objective optimal value for $\mathcal{M}$. $\hat{\pi}^*$ refers to the optimal policy in the approximate IALM $\hat{\mathcal{M}}$. $\mathcal{V}^{\hat{\pi}^*}$ denotes the value achieved by policy $\hat{\pi}^*$ in the IALM $\mathcal{M}$. Our aim is to bound the value loss given by the difference $\mathcal{V}^* - \mathcal{V}^{\hat{\pi}^*}$. We define $|R| \triangleq \max_{a,s} |R(s, a)|$. We use $|| \, . \, ||_1$ and D_{KL} to denote the 1-norm and the KL divergence between probability distributions.

The proof of the value loss bound derived is divided into two steps. We proceed first by bounding the value loss with the difference between the IALM transitions functions $||\mathcal{T} - \hat{\mathcal{T}}||_1$. Then, we prove an upper bound for $||\mathcal{T} - \hat{\mathcal{T}}||_1$ in terms of the difference $||I - \hat{I}||_1$. Finally, combining these two results we derive an upper bound for the value loss in terms of the distance between the EIP and the AIP $||I - \hat{I}||_1$. We collect the proofs of all the following claims in the appendix of the extended version of this paper [13].

Since the IALM $\hat{\mathcal{M}}$ is essentially an approximation of the exact influence IALM $\mathcal{M}$ with imprecise transitions, we apply the loss bounds for MDPs with uncertain transition probabilities [16, 24, 37] to the case of the IALMs.

Theorem 1. Consider the two IALMs $\mathcal{M}$ and $\hat{\mathcal{M}}$, the following loss bound holds:

$$||\mathcal{V}^* - \mathcal{V}^{\hat{\pi}^*}||_\infty \leq 2h^2|R| \max_t \max_{\bar{s}^t, a^t} ||(\mathcal{T}(\cdot|\bar{s}^t, a^t) - \hat{\mathcal{T}}(\cdot|\bar{s}^t, a^t)||_1 \quad (2)$$

The importance is that we can bound the value loss when the IALM transition functions $\mathcal{T}$, $\hat{\mathcal{T}}$ induced by I, $\hat{I}$ are sufficiently close.

For the second step, we use the relation between the IALM transitions $\mathcal{T}$ and the influence point I expressed by 1.

Lemma 1. Consider the IALMs transitions $\mathcal{T}$ and $\hat{\mathcal{T}}$. For any t, augmented state $\bar{s}^t = (x^t, d^t)$ and action a^t,

$$||(\mathcal{T}(\cdot|\bar{s}^t, a^t) - \hat{\mathcal{T}}(\cdot|\bar{s}^t, a^t)||_1 \leq ||(I^t(\cdot|d^t) - \hat{I}^t(\cdot|d^t)||_1 \quad (3)$$

Combining Theorem 1 and Lemma 1, we obtain the L^1 loss bound.

Theorem 2. Consider an IALM $\mathcal{M} = (\mathcal{S}, A, \mathcal{T}, R, h, b^0)$ and an AIP $\hat{I}$ inducing $\hat{\mathcal{M}} = (\mathcal{S}, A, \hat{\mathcal{T}}, R, h, b^0)$. Then, a value loss bound in terms of the 1-norm error is given by

$$||\mathcal{V}^* - \mathcal{V}^{\hat{\pi}^*}||_\infty \leq 2h^2|R| \max_{t, d^t} ||(I^t(\cdot|d^t) - \hat{I}^t(\cdot|d^t)||_1 \quad (4)$$

This shows that the value loss is bounded by the worst case L^1-distance between the exact and the approximate influence over all the possible d-set instantiations.

5.2 KL Divergence Bound

We now present a loss bound in terms of the KL divergence between the approximate and the exact influence. Since the cross entropy and KL divergence differ solely by an additive constant, this result establishes a relation between the value loss and the cross entropy error for influence approximation.

Corollary 1. Consider an IALM $\mathcal{M} = (\mathcal{S}, A, \mathcal{T}, R, h, b^0)$ and an AIP $\hat{I}$ inducing $\hat{\mathcal{M}} = (\mathcal{S}, A, \hat{\mathcal{T}}, R, h, b^0)$. Then, a value loss bound in terms of KL divergence error is given by

$$||\mathcal{V}^* - \mathcal{V}^{\hat{\pi}^*}||_\infty \leq 2h^2|R| \max_{t, d^t} \sqrt{2D_{KL}(I^t(\cdot|d^t)||\hat{I}^t(\cdot|d^t))} \quad (5)$$

We see that the performance loss can be bounded by the max KL divergence. Cross entropy loss optimizes the mean of such loss $\mathbb{E}_{t, D^t}\left[D_{KL}(I^t(\cdot|D^t))||\hat{I}^t(\cdot|D^t)\right]$ and therefore is aligned with this bound: even though it does not optimize the max itself, it is intuitively clear that in many problems a low mean will imply a low max error, even the more since the mean is known to be sensitive

to outliers. Moreover, this is an asymptotically tight bound. That is, when the KL divergence tends to zero the approximate influence solution approaches the true optimal one. This suggests that neural networks learning approaches are well suited for the approximate influence learning task and that the cross entropy, which has been widely used as test error, can give a priori insight on the value loss.

5.3 Probabilistic Loss Bound

The previous section gives an idea of what AIPs properties can guarantee bounded value loss. However, the direct application of Theorem 2 and Corollary 1 requires the distance to the true influence point I, which is unknown. In this section, we present an approach to overcome this limitation: a probabilistic loss bound depending only on the distance to an empirical influence distribution, which can be measured using a test set.

Precisely, assume that for a given instatiation d^t of the d-set at time t we have N samples of the influence sources $\left\{y_{\text{src},1}^t, \ldots, y_{\text{src},N}^t\right\}$ and let

$$I_N^t \left(y_{\text{src}}^t \mid d^t \right) = \frac{1}{N} \sum_{k=1}^N \delta_{y_{\text{src},k}^t} (y_{\text{src}}^t)$$

denote the empirical conditional distribution. Our result presents a value loss bound in terms of the distance between the empirical influence I_N^t and the approximate $\hat{I}^t$.

Theorem 3. Consider an IALM $\mathcal{M} = (\mathcal{S}, A, \mathcal{T}, R, h, b^0)$ and an AIP $\hat{I}$ inducing $\hat{\mathcal{M}} = (\mathcal{S}, A, \hat{\mathcal{T}}, R, h, b^0)$. Assume that for every time t and d-set instantiation d^t, we have at least an sample of size N of influence sources. Then for every $\varepsilon > 0$

$$\mathbb{P}\left(||\mathcal{V}^* - \mathcal{V}^{\hat{\pi}^*}||_\infty \leq 2h^2 |R| \max_{t,d^t} ||(I_N^t(\cdot \mid d^t) - \hat{I}^t(\cdot \mid d^t)||_1 + \varepsilon\right) \geq$$

$$1 - h|D^h|(2^{|Y_{\text{src}}|} - 2)e^{-N\frac{\varepsilon^2}{2}} \tag{6}$$

where $|Y_{\text{src}}|$ is the cardinality of the influence sources space and $|D^h|$ the cardinality of the d-sets space at time h.

This theorem states that, with high probability, the value loss of using an AIP $\hat{I}$ is upper bounded by the empirical error (that we can measure using a test set). We believe that this is a first step towards bounds that can be used in practical applications.

However, the bound in Theorem 3, still requires an intractable maximization over all possible instantiations of d-sets (and needs $N > \log(h|D^h|2^{|Y_{\text{src}}|})$ samples for each of these). Moreover, the maximization over d-sets might be too conservative in many cases. Namely, the right-hand side of (5) corresponds to the loss one agent would incur if every time step the mistake is caused by the worst-case approximation of the influence over d-sets. We think that the maximization may be replaced with an appropriate expectation over instantiations. In this respect, identifying versions of these bounds that work with expectations over d-sets, rather than maximization, is one of the important directions of future work that our paper identifies.

6 MULTIAGENT IMPLICATIONS

So far, we have focused on the perspective of a single agent, that computes a best response against the fixed policies of other agents. However, the results in this paper can have important implications

also for settings where we optimize or solve for the policies of multiple agents at the same time. In fact, the insights of influence-based abstraction originated from the study of multiagent systems [4, 5, 32, 36, 38–44], so these implications should hardly be surprising. Nevertheless, we think it is useful to spell out some of these implications and will do so in the remainder of this section.

First, we point out that the best-response setting that we consider is very general. In fact, the notion of a 'fixed policy' of an agent in a POSG is very powerful: such a fixed policy is a mapping from histories of actions and observations to distributions over actions and therefore powerful enough to model learning agents. This means that the fixed policies that we compute a best response against could include, e.g., a Q-learning agent.

A second implication is for the computation of equilibria in POSGs. Many methods for computing Nash equilibria in POSGs [6, 11, 21, 25, 31], use best-response computation as an inner loop. In many cases, such as in the double-oracle algorithm [25] or the parallel Nash Memory [31], replacing such a best-response computation with an ϵ-best response computation can enable us to compute ϵ-approximate Nash Equilibria (ϵ-NEs) [27], in which no player can benefit more than ϵ from deviating. As such, our results – that show under what conditions on the influence predictions we can compute an ϵ-approximate best response – may lead to computationally feasible paths to compute ϵ-NEs.

Even in cases where computing ϵ-NEs will remain out of reach, our results can provide insight in complex MAS.

Observation 1. Given a method E to estimate an AIP with an error (maximum L_1 norm) of at most ϵ_1, we can verify if a particular joint policy $\pi = (\pi_1, \ldots, \pi_n)$ is an $(2h^2|R|\epsilon_1)$-NE as follows: 1) use E to estimate the influence $\hat{I}_i$ on each agent i 2) verifying if π_i is an optimal solution in the IALM constructed for agent i.

Of course, the question of how to develop such estimators E is not trivial in the general case, but there are many special cases with compact influence descriptions where this is possible [12, 33].

Moreover, IBA serves as the basis for *influence search* [5, 42, 44] in cooperative settings. The key idea is that an influence point captures all the relevant information about many different policies of the other agents. That is, different joint policies might induce the same influences on the local models of the agents. Thus, the space of 'joint influence points' $I = (I_1, \ldots, I_n)$ can be much smaller than the space of joint policies $\pi = (\pi_1, \ldots, \pi_n)$, and it can be much more efficient to search through the former. [43].[1]

7 EMPIRICAL EVALUATION

The bounds presented in Section 5 establish a relation between the value loss and the cross entropy of the influence approximations. In particular, they suggest that the *mean* cross entropy loss is aligned with the objective of minimizing the performance loss.

Here we evaluate if this alignment translates into practical settings by investigating the relation between the mean cross entropy

$$\text{CE}(I, \hat{I}) \triangleq \mathbb{E}_{t,D^t} \left[\text{CE}(I^t(\cdot|D^t), \hat{I}^t(\cdot|D^t)) \right]$$

---

[1]Roughly speaking, an influence search algorithm searches in the space of possible joint influences, and for each joint influence point $I = (I_1, \ldots, I_n)$, it solves a local constrained best-response problem for each agent (possibly in parallel): each agent i computes a best-response $\hat{\pi}^*(I)$ to I_i constrained to inducing influences I_{-i} on the local models of the other agents.
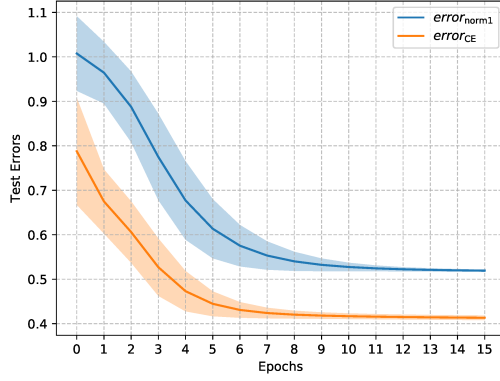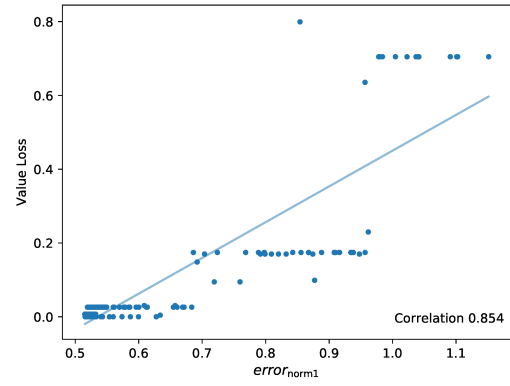
(a) Test errors for increasing training epochs.
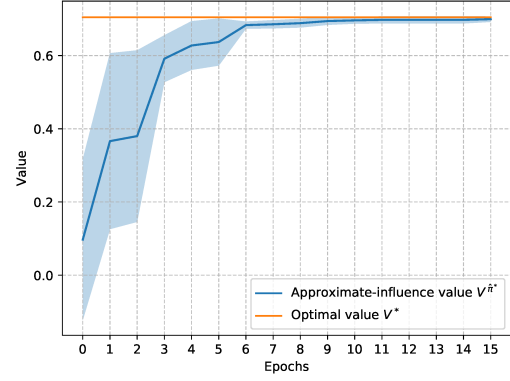
(b) Value achieved by $\hat{\pi}^*$ compared to the optimal value for increasing training epochs.
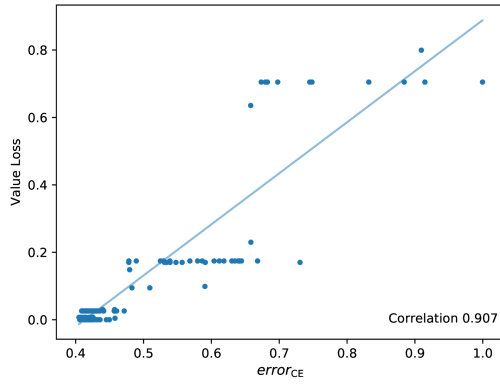
(c) Correlation between value loss and $error_{CE}$.

(d) Correlation between value loss and $error_{norm1}$.

Figure 2: Planetary Exploration.

and the performance loss $\mathcal{V}^* - \mathcal{V}^{\hat{\pi}^*}$. Namely, we expect to see that a better influence approximation in terms of cross entropy loss, indeed, leads to a smaller value loss. Here we try to validate this hypothesis.

Similarly, we investigate to what extent the *mean* 1-norm error

$$||I - \hat{I}||_1 \triangleq \mathbb{E}_{t,D^t} \left[||I^t(\cdot|D^t) - \hat{I}^t(\cdot|D^t)||_1 \right]$$

allows to a priori assess the quality of the approximations in terms of the value loss. In fact, according to the results in Section 5, the 1-norm has the potential to provide a tighter bound for the performance loss. If experimentally confirmed, this would motivate future work to investigate the possibility to use different training losses based on L^1 distance.

7.1 Experimental Setup

We follow the procedure sketched in Algorithm 1. That is, we first run the simulations from the global model using an exploratory random policy π^{Exp} for the local agent and the set of fixed policies for the other agents. We collect the samples of influence sources and
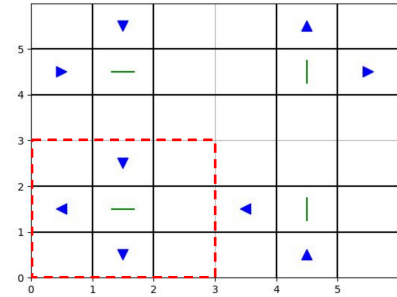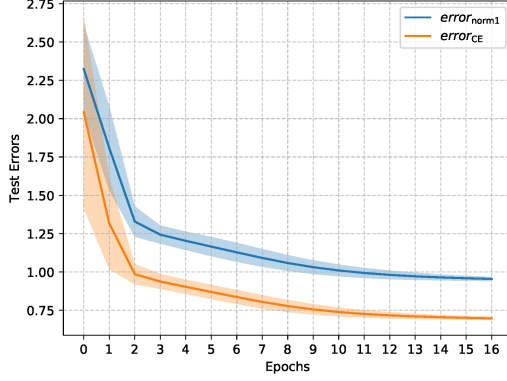

Figure 3: Traffic network. The local model is delimited by the dotted square. The blue triangles represent the vehicles and the green bars the traffic lights.
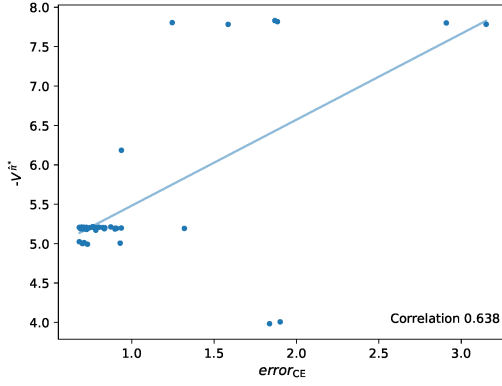
d-sets $\{y^t_{\mathrm{src,k}}, d^t_k\}_{t=1:h,k=1:N}$. We train a LSTM neural network with cross entropy loss to induce the approximate influence $\hat{I}^t(\cdot|D^t)$. At different training epochs, for the corresponding approximations of
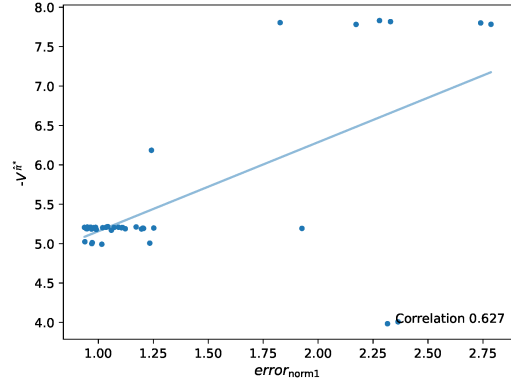
(a) Test errors for increasing training epochs.
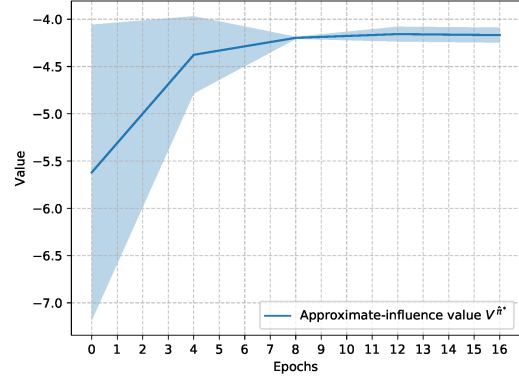
(b) Value achieved by $\hat{\pi}^*$ for increasing training epochs computed every 4 epochs.

(c) Correlation between value loss and $error_{CE}$.

(d) Correlation between value loss and $error_{norm1}$.

Figure 4: Traffic network.

the influence, we build the approximate-influence local model. Then, we compute the optimal policy $\hat{\pi}^*$ in the approximate-influence model through value iteration [8]. We evaluate the policy $\hat{\pi}^*$ in the global model using M simulations to obtain the value achieved $\mathcal{V}^{\hat{\pi}^*}$. Note that since we use an exact solution method to compute the policy $\hat{\pi}^*$ in the approximate-influence model, any other solution method would get lower performance in the local model. For each epoch, to assess the quality of the approximation, we use Monte Carlo estimators for the mean cross entropy $CE(I, \hat{I})$ and the 1-norm error $||I - \hat{I}||_1$, computed using a test set as

$$error_{CE} = -\frac{1}{h}\sum_{t=1}^{h}\frac{1}{N}\sum_{k=1}^{N}\ln\left(\hat{I}^t(y_{src,k}^t \mid d_k^t)\right)$$

$$error_{norm1} = \frac{1}{h}\sum_{t=1}^{h}\frac{1}{N}\sum_{k=1}^{N}\left\|\delta_{y_{src,k}^t}(\cdot) - \hat{I}^t(\cdot \mid d_k^t)\right\|_1 \quad (7)$$

We compute the average and standard deviation of the $error_{CE}$ and $error_{norm1}$ over different iterations of the same experiment. For any iteration, we repeat entirely the steps described above: we

recollect training and testing samples from the global model, retrain the neural network etc. When the problem is sufficiently easy to solve, we compute the optimal value $\mathcal{V}^*$ and then we measure the correlation between $error_{CE}$, $error_{norm1}$ and the value loss $\mathcal{V}^* - \mathcal{V}^{\hat{\pi}^*}$. Otherwise, we measure the correlation between the test errors and $-\mathcal{V}^{\hat{\pi}^*}$. In fact, since the optimal value $\mathcal{V}^*$ is constant for increasing epochs, it does not affect the correlation. That is, $Corr(error, \mathcal{V}^* - \mathcal{V}^{\hat{\pi}^*}) = Corr(error, -\mathcal{V}^{\hat{\pi}^*})$.

We run the experiments in three domains: a version of the planetary exploration [42], a traffic domain and the fire fighters problem [28]. Section 3.3 provides the overview of the planetary exploration domain and high level descriptions of the other domains follow. For more details we refer to the appendix [13]. Our codebase is available at https://github.com/INFLUENCEorg/Approx-IBA-Planning.

7.1.1 Traffic Network. In this domain, we simulate a traffic network with 4 intersections (see Figure 3). The sensors of the traffic lights at each intersection provide information on the 3×3 local grid around them. In Figure 3, the dotted red square represents the local model for the protagonist agent. The other traffic lights employ

hand-coded policies prioritizing fixed lanes. The goal of the agent is to minimize the total number of vehicles waiting at the local intersection. In order to act optimally, the local agent only needs to predict if there will be incoming cars from the *east* and *north* lanes of the local model the next time step. The outgoing cars have some probability to re-enter in the network from other lanes. That is, the outgoing vehicles from *south* and *west* can affect the decisions of other traffic lights and consequently, the vehicles inflow in the local model at future time steps.

7.1.2 Fire fighters. We model a team of 2 agents that need to co-operate to extinguish fires in a row of 3 houses. At every time step, every agent can choose to fight fires at one of its 2 neighboring houses. The goal of each agent is to minimize the number of neighboring houses that are burning. We take the perspective of one agent with a local model including only the two neighboring houses.

7.2 Experimental Results

Figures 2(a), 4(a), 5(a) show the test errors measured by $error_{CE}$ and $error_{norm1}$ 7, as functions of the training epochs for the three domains. They show that performance in terms of cross entropy and 1-norm test errors improves monotonically with the number of epochs. These trends suggest that the AIP improves with training, the question is if this also corresponds to a value loss decrease. Looking at Figure 2(b) this seems to be the case in the planetary exploration setting. We see that the performance of the policy derived from the approximate-influence IALM is very close to optimal from 6 epochs onward, right about when the training of the neural network starts to stagnate. Moreover, Figure 2(c) and 2(d) show that the decrease in mean empirical cross entropy or 1-norm error indeed correlates well with actual value loss. In the traffic domain, we also see that there is a significant improvement in the value achieved by the approximate-influence policy $\hat{\pi}^*$ for increasing number of epochs in Figure 4(b). The test errors seem still well aligned with the value loss in Figures 4(c), 4(d). For the fire fighters domain, Figure 5(b) shows that the value improves over training epochs coherently with the decrease of the test errors.

For more results on correlation analysis on different settings of the planetary exploration, see the Supplementary Material of the extended version of this paper [13].

We can conclude that both the empirical cross entropy error and the 1-norm errors correlate well with the value loss and therefore provide a priori insight on the quality of the influence-approximation in terms of the value achieved.

8 CONCLUSIONS AND DISCUSSION

In this paper, we provided a priori quality guarantees for the value loss resulting from approximating the influence point. Our results show that the objective of a neural network trained with the cross entropy loss is well aligned with the goal of minimizing the value loss. Our empirical results demonstrate that empirical (mean) estimators of the cross entropy and the 1-norm are predictive of the value loss. We used this insight to evaluate the transfer of our theoretical results to practical scenarios.

In this work, we limit the discussion to fully observable local models. However, in the general case, we would be dealing with a
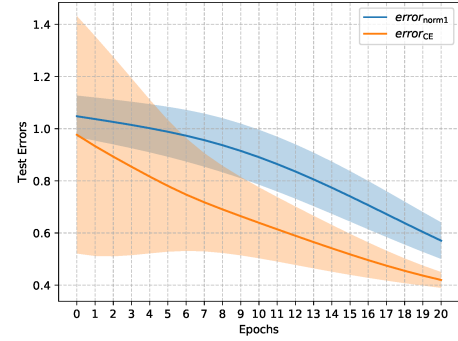
(a) Test errors for increasing training epochs.
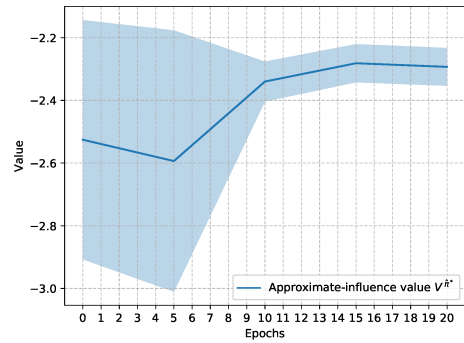
(b) Value achieved by $\hat{\pi}^*$ for increasing training epochs computed every 5 epochs.

Figure 5: Fire Fighters Problem.

local POMDP. We believe that a generalization of our bound in the partially observable case can be obtained for instance leveraging the 'back-projected value vectors' formulation [32]. As future work, we intend to further investigate this.

The current formulation of the loss bounds involves a maximization over the space of all the possible instantiations of the d-set, which might grow exponentially in time. In future work, we would like to derive expressions that do not need a maximization, but for example are in expectation over a sampling process. This would lead to tighter bounds, which could be used in practice for model selection or value loss estimation.

ACKNOWLEDGMENTS

REFERENCES

[1] David Abel, David Hershkowitz, and Michael Littman. 2016. Near Optimal Behavior via Approximate State Abstraction. In *Proc. of the International Conference on Machine Learning*. 2915–2923.

[2] Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2016. An Actor-Critic Algorithm for Sequence Prediction. *arXiv preprint arXiv:1607.07086* (2016).

[3] Aijun Bai, Siddharth Srivastava, and Stuart J Russell. 2016. Markovian State and Action Abstractions for MDPs via Hierarchical MCTS. In *Proc. of the International Joint Conference on Artificial Intelligence*. 3029–3039.

[4] Raphen Becker, Shlomo Zilberstein, and Victor Lesser. 2004. Decentralized Markov Decision Processes with Event-Driven Interactions. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems*. 302–309.

[5] Raphen Becker, Shlomo Zilberstein, Victor Lesser, and Claudia V Goldman. 2004. Solving Transition Independent Decentralized Markov Decision Processes. *Journal of Artificial Intelligence Research* 22 (2004), 423–455.

[6] Ulrich Berger. 2007. Brown's Original Fictitious Play. *Journal of Economic Theory* 135, 1 (2007), 572–578.

[7] Daniel S. Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. 2002. The Complexity of Decentralized Control of Markov Decision Processes. *Mathematics of Operations Research* 27, 4 (2002), 819–840.

[8] Dimitri P Bertsekas. 1995. *Dynamic Programming and Optimal Control*. Vol. 1. Athena scientific Belmont, MA.

[9] Christopher M Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer.

[10] Craig Boutilier, Thomas Dean, and Steve Hanks. 1999. Decision-theoretic planning: structural assumptions and computational leverage. *Journal of Artificial Intelligence Research* 11 (1999), 1–94.

[11] George W Brown. 1951. Iterative Solution of Games by Fictitious Play. *Activity analysis of production and allocation* 13, 1 (1951), 374–376.

[12] Rohan Chitnis and Tomás Lozano-Pérez. 2020. Learning Compact Models for Planning with Exogenous Processes. In *PMLR*. 813–822.

[13] Elena Congeduti, Alexander Mey, and Frans A Oliehoek. 2020. Loss Bounds for Approximate Influence-Based Abstraction. *arXiv preprint arXiv:2011.01788* (2020).

[14] Thomas L Dean, Robert Givan, and Sonia Leach. 2013. Model reduction techniques for computing approximately optimal solutions for Markov decision processes. *arXiv preprint arXiv:1302.1533* (2013).

[15] Richard Dearden and Craig Boutilier. 1997. Abstraction and approximate decision-theoretic planning. *Artificial Intelligence* 89, 1-2 (1997), 219–283.

[16] Karina Valdivia Delgado, Scott Sanner, and Leliane Nunes De Barros. 2011. Efficient solutions to factored MDPs with imprecise transition probabilities. *Artificial Intelligence* 175, 9-10 (2011), 1498–1527.

[17] Robert Givan, Sonia Leach, and Thomas Dean. 2000. Bounded-parameter Markov decision processes. *Artificial Intelligence* 122, 1-2 (2000), 71–109.

[18] Claudia V Goldman and Shlomo Zilberstein. 2004. Decentralized Control of Cooperative Systems: Categorization and Complexity Analysis. *Journal of Artificial Intelligence Research* 22 (2004), 143–174.

[19] Eric A Hansen, Daniel S Bernstein, and Shlomo Zilberstein. 2004. Dynamic programming for partially observable stochastic games. In *Proc. of the AAAI Conference on Artificial Intelligence*. 709–715.

[20] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Shun Chen. 2017. LSTM fully convolutional networks for time series classification. *IEEE access* 6 (2017), 1662–1669.

[21] Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Julien Perolat, David Silver, and Thore Graepel. 2017. A unified game-theoretic approach to multiagent reinforcement learning. In *Proc. of the Advances in Neural Information Processing Systems*.

[22] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. 2017. Temporal convolutional networks for action segmentation and detection. In *In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*. 156–165.

[23] Lihong Li, Thomas J Walsh, and Michael L Littman. 2006. Towards a Unified Theory of State Abstraction for MDPs. In *In Proc. of the International Symposium on Artificial Intelligence and Mathematics*.

[24] Andrew Mastin and Patrick Jaillet. 2012. Loss bounds for uncertain transition probabilities in Markov decision processes. In *Proc. of the IEEE Conference on Decision and Control*. IEEE, 6708–6715.

[25] H Brendan McMahan, Geoffrey J Gordon, and Avrim Blum. 2003. Planning in the presence of cost functions controlled by an adversary. In *Proc. of the International Conference on Machine Learning*. 536–543.

[26] Ranjit Nair, Milind Tambe, Makoto Yokoo, David V. Pynadath, and Stacy Marsella. 2003. Taming Decentralized POMDPs: Towards Efficient Policy Computation for Multiagent Settings. In *Proc. of the International Joint Conference on Artificial Intelligence*. 705–711.

[27] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani. 2007. *Algorithmic Game Theory*. Cambridge University Press.

[28] Frans A Oliehoek. 2010. *Value-based planning for teams of agents in stochastic partially observable environments*. Amsterdam University Press.

[29] Frans A Oliehoek and Christopher Amato. 2016. *A Concise Introduction to Decentralized POMDPs*. Springer.

[30] Frans A Oliehoek, Christopher Amato, et al. 2014. Best response Bayesian reinforcement learning for multiagent systems with state uncertainty. In *AAMAS Workshop on Multiagent Sequential Decision Making Under Uncertainty*.

[31] Frans A Oliehoek, Edwin D De Jong, and Nikos Vlassis. 2006. The parallel Nash memory for asymmetric games. In *Proc. of the conference on Genetic and evolutionary computation*. 337–344.

[32] Frans A Oliehoek, Matthijs T J Spaan, and Stefan J Witwicki. 2015. Factored Upper Bounds for Multiagent Planning Problems under Uncertainty with Non-Factored Value Functions. In *Proc. of the International Joint Conference on Artificial Intelligence*. 1645–1651.

[33] Frans A Oliehoek, Stefan J Witwicki, and Leslie P Kaelbling. 2019. A Sufficient Statistic for Influence in Structured Multiagent Environments. *arXiv preprint arXiv:1907.09278* (2019).

[34] Alessandro Panella and Piotr Gmytrasiewicz. 2017. Interactive POMDPs with finite-state models of other agents. *Journal of Autonomous Agents and Multi-Agent Systems* 31, 4 (01 July 2017), 861–904.

[35] Marek Petrik and Dharmashankar Subramanian. 2014. RAAM: The benefits of robustness in approximating aggregated MDPs in reinforcement learning. In *Advances in Neural Information Processing Systems*. 1979–1987.

[36] Marek Petrik and Shlomo Zilberstein. 2009. A Bilinear Programming Approach for Multiagent Planning. *Journal of Artificial Intelligence Research* 35, 1 (2009), 235–274.

[37] Alexander L Strehl, Lihong Li, and Michael L Littman. 2009. Reinforcement learning in finite MDPs: PAC analysis. *Journal of Machine Learning Research* 10 (2009), 2413–2444.

[38] Pradeep Varakantham, Jun-young Kwak, Matthew E Taylor, Janusz Marecki, Paul Scerri, and Milind Tambe. 2009. Exploiting Coordination Locales in Distributed POMDPs via Social Model Shaping. In *Proc. of the International Conference on Automated Planning and Scheduling*.

[39] Prasanna Velagapudi, Pradeep Varakantham, Paul Scerri, and Katia Sycara. 2011. Distributed Model Shaping for Scaling to Decentralized POMDPs with hundreds of agents. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems*.

[40] Stefan J Witwicki and Edmund Durfee. 2010. From Policies to Influences: a Framework for Nonlocal Abstraction in Transition-dependent Dec-POMDP Agents. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems*. 1397–1398.

[41] Stefan J Witwicki and Edmund H Durfee. 2009. Flexible Approximation of Structured Interactions in Decentralized Markov Decision Processes. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems*. 1251–1252.

[42] Stefan J Witwicki and Edmund H Durfee. 2010. Influence-Based Policy Abstraction for Weakly-Coupled Dec-POMDPs. In *Proc. of the International Conference on Automated Planning and Scheduling*.

[43] Stefan J Witwicki and Edmund H Durfee. 2011. Towards a Unifying Characterization for Quantifying Weak Coupling in Dec-POMDPs. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems*. 29–36.

[44] Stefan J Witwicki, Frans A Oliehoek, and Leslie P Kaelbling. 2012. Heuristic Search of Multiagent Influence Space. In *Proc. of the International Conference on Autonomous Agents and Multiagent Systems*. 973–981.