Online Inverse Reinforcement Learning Under Occlusion

Saurabh Arora, Prashant Doshi
THINC Lab, Dept. of Computer Science
University of Georgia, Athens, GA
{sa08751,pdoshi}@uga.edu

Bikramjit Banerjee
School of Computing Sciences & Computer Engineering
University of Southern Mississippi, Hattiesburg, MS
Bikramjit.Banerjee@usm.edu

ABSTRACT

Inverse reinforcement learning (IRL) is the problem of learning the preferences of an agent from observing its behavior on a task. While this problem is witnessing sustained attention, the related problem of *online* IRL – where the observations are incrementally accrued, yet the real-time demands of the application often prohibit a full rerun of an IRL method – has received much less attention. We introduce a formal framework for online IRL, called *incremental IRL* (I2RL), and a new method that advances maximum entropy IRL with hidden variables, to this setting. Our analysis shows that the new method has a monotonically improving performance with more demonstration data, as well as probabilistically bounded error, both under full and partial observability. Experiments in a simulated robotic application, which involves learning under occlusion, show the significantly improved performance of I2RL as compared to both batch IRL and an online imitation learning method.

KEYWORDS

Robot Learning; Online Learning; Robotics; Reinforcement Learning; Inverse Reinforcement Learning

ACM Reference Format:
Saurabh Arora, Prashant Doshi and Bikramjit Banerjee. 2019. Online Inverse Reinforcement Learning Under Occlusion. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 9 pages.

1 INTRODUCTION

Inverse reinforcement learning (IRL) [13, 17] refers to the problem of ascertaining an agent's preferences from observations of its behavior while executing a task. It inverts RL with its focus on learning the reward function that explains the input behavior. IRL lends itself naturally to learning from demonstrations in controlled environments, and therefore finds application in robot learning from demonstration by a human teacher [2], imitation learning [14], and in ad hoc collaborations [19].

Previous methods for IRL [1, 3, 8, 9, 15] typically operate on large batches of observations and yield an estimate of the expert's reward function in a one-shot manner. These methods fill the need of applications that predominantly center on imitation learning. Here, the task being performed is observed and must be replicated subsequently. However, newer applications of IRL are motivating the need for continuous learning from streaming data or data in mini-batches. Consider, for example, the task of forecasting a person's goals in an everyday setting from observing her ongoing activities using a body camera [16]. Alternately, a robotic learner observing continuous patrols from a vantage point is tasked with penetrating the patrolling and reaching a goal location speedily and without being spotted [4]. Both these applications offer streaming observations, and would benefit from progressively learning and assessing expert's preferences.

In this paper, we present a formal framework to facilitate investigations into *online* IRL. The framework, labeled as incremental IRL (I2RL), establishes the key components of this problem and rigorously defines the notion of an incremental variant of IRL. Jin et al. [12] and Rhinehart et al. [16] introduced IRL methods that are suited for online IRL, and we cast these in the context provided by I2RL. Next, we introduce a new method that generalizes recent pragmatic advances in maximum entropy IRL with partially hidden demonstration data [7] to an online setting. Key theoretical properties of this new method are also established.

Our experiments evaluate the benefit of online IRL on the previously introduced robotic application of IRL toward penetrating continuous patrols under occlusion [4]. We comprehensively demonstrate that the new incremental method achieves a reasonably good learning performance that is similar to that of the previously introduced batch method in significantly less time. Thus, it suffers from far fewer timeouts (a timeout occurs when learning and planning is not completed within an imposed time-limit) and admits a significantly improved success rate. Given the partially occluded trajectory data, our method also learned more accurately than a leading online imitation learning method that uses generative adversarial networks [11]. Consequently, this paper makes important initial contributions toward the nascent problem of online IRL by offering both a formal framework, I2RL, and a new general method that has convergence guarantees and performs well.

2 BACKGROUND ON IRL

Informally, IRL refers to both the problem and method by which an agent learns preferences of another agent that explain the latter's observed behavior [17]. Usually considered an "expert" in the task that it is performing, the observed agent, say E, is modeled as executing the optimal policy of a standard MDP defined as $\langle S_E, A_E, T_E, R_E \rangle$. The learning agent L is assumed to perfectly know the parameters of the MDP except the reward function. Consequently, the learner's task may be viewed as finding a reward function under which the expert's observed behavior is optimal.

This problem, in general, is ill-posed because for any given behavior there are infinitely-many reward functions which align with the behavior. Ng and Russell [13] first formalized this task as a linear program in which the reward function that maximizes the difference in value between the expert's policy and the next best policy is sought. Abbeel and Ng [1] present an algorithm that allows the expert E to provide task demonstrations instead of its

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada. © 2019 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

policy. The reward function is modeled as a linear combination of K binary features, ϕ_k: $S_E \times A_E \rightarrow [0,1]$, $k \in \{1, 2 \ldots K\}$, each of which maps a state from the set of states S_E and an action from the set of E's actions A_E to a value in [0,1]. Note that non-binary feature functions can always be converted into binary feature functions although there will be more of them. Throughout this article, we assume that these features are known to or selected by the learner. The reward function for expert E is then defined as $R_E(s,a) = \boldsymbol{\theta}^T \phi(s,a) = \sum_{k=1}^{K} \theta_k \cdot \phi_k(s,a)$, where θ_k are the *weights* in vector $\boldsymbol{\theta}$; let $\mathcal{R} = \mathbb{R}^{|S_E \times A_E|}$ be the continuous space of the reward functions. The learner's task is reduced to finding a vector of weights that complete the reward function, and subsequently, the MDP such that the demonstrated behavior is optimal. Let $\mathbb{N}^+$ be a bounded set of natural numbers.

Definition 1 (Set of fixed-length trajectories). *The set of all trajectories of finite length T from an MDP attributed to the expert E is defined as,* $\mathbb{X}^T = \{X | X = (\langle s,a\rangle_1, \langle s,a\rangle_2, \ldots, \langle s,a\rangle_T), T \in \mathbb{N}^+\}, \forall s \in S_E, \forall a \in A_E\}$.

Then, the set of *all* trajectories is $\mathbb{X} = \mathbb{X}^1 \cup \mathbb{X}^2 \cup \ldots \cup \mathbb{X}^{|\mathbb{N}^+|}$. A demonstration is some finite set of trajectories of varying lengths, $\mathcal{X} = \{X^T | X^T \in \mathbb{X}^T, T \in \mathbb{N}^+\}$, and it includes the empty set. [1] Subsequently, we may define the set of demonstrations.

Definition 2 (Set of demonstrations). *The set of demonstrations is the set of all subsets of the space of trajectories of varying lengths. Therefore, it is the power set,* $2^{\mathbb{X}} = 2^{\mathbb{X}^1 \cup \mathbb{X}^2 \cup \ldots \cup \mathbb{X}^{|\mathbb{N}^+|}}$.

In the context of the definitions above, traditional IRL attributes an MDP without the reward function to the expert, and usually involves determining an estimate of the expert's reward function, $\hat{R}_E \in \mathcal{R}$, which best explains the observed demonstration, $\mathcal{X} \in 2^{\mathbb{X}}$. As such, we may view IRL as a function: $\zeta(MDP_{/R_E}, \mathcal{X}) = \hat{R}_E$.

To assist in finding the weights, feature expectations for the expert's demonstration are empirically estimated and compared to those of all possible trajectories [22]. Feature expectations of the expert are estimated as a discounted average over feature values for all observed trajectories, $\hat{\phi}_k = \frac{1}{|X|} \sum_{X \in \mathcal{X}} \sum_{\langle s,a\rangle_t \in X} \gamma^t\ \phi_k(\langle s,a\rangle_t)$, where X is a trajectory in the set of all observed trajectories, $\mathcal{X}$, and $\gamma \in (0,1)$ is a discount factor. After learning a set of reward weights, expert's MDP is completed and solved optimally to produce π_E. The difference $\hat{\phi} - \phi^{\pi_E}$ provides a gradient with respect to the reward weights for a numerical solver.

2.1 Maximum Entropy IRL

While expected to be valid in some contexts, the max-margin approach of Abeel and Ng [1] introduces a bias into the learned reward function in general. To address this, Ziebart et al. [22] find the distribution with maximum entropy over all trajectories that is constrained to match the observed feature expectations.

$$\begin{array}{l} \max\limits_{\Delta} \left(-\sum_{X \in \mathbb{X}} P(X;\boldsymbol{\theta})\ log\ P(X;\boldsymbol{\theta})\right) \\ \textbf{subject to } \sum_{X \in \mathbb{X}} P(X;\boldsymbol{\theta}) = 1 \\ E_{\mathbb{X}}[\phi_k] = \hat{\phi}_k\ \forall k \end{array} \tag{1}$$

Here, Δ is the space of all distributions over the space $\mathbb{X}$ of all trajectories, and $E_{\mathbb{X}}[\phi_k] = \sum_{X \in \mathbb{X}} P(X)\ \sum_{\langle s,a\rangle_t \in X} \gamma^t \phi_k(\langle s,a\rangle_t)$. As the distribution P(·) is parameterized by learned weights $\boldsymbol{\theta}$, $E_{\mathbb{X}}[\phi_k]$ represents the feature expectations $\phi_k^{\pi_E}$. The benefit is that distribution $P(X;\boldsymbol{\theta})$ makes no further assumptions beyond those which are needed to match its constraints and is maximally noncommittal to any one trajectory. As such, it is most generalizable by being the least wrong most often of all alternative distributions. A disadvantage is that it becomes intractable for long trajectories because the set of trajectories grows exponentially with length. In this regard, another formulation defines the maximum entropy distribution over policies [8], the size of which is also large but fixed.

2.2 IRL under Occlusion

Our motivating application involves a subject robot that must observe other mobile robots from a fixed vantage point. Its local sensors allow it a limited observation area; within this area, it can observe the other robots fully, outside this area it cannot observe at all. Previous methods [4, 5] denote this special case of partial observability where certain states are either fully observable or fully hidden as *occlusion.* Subsequently, the trajectories gathered by the learner exhibit missing data associated with time steps where the expert robot is in one of the occluded states. The empirical feature expectation of the expert $\hat{\phi}_k$ will thus exclude the occluded states (and actions in those states).

Bogert and Doshi [4], while maximizing entropy over policies [8], limited the calculation of feature expectations for policies to observable states only. To ensure that the feature expectation constraint in IRL accounts for the missing data, a recent approach [6, 7] by same authors improves on this method by taking an expectation over the missing data conditioned on the observations. Completing the missing data in this way allows the use of all states in the constraint and with it the Lagrangian dual's gradient as well. The nonlinear program in (1) is modified to account for the hidden data and its expectation.

Let Y be the observed portion of a trajectory, Z is one way of completing the hidden portions of this trajectory, and $X = Y \cup Z$. Now we may treat Z as a latent variable and take the expectation to arrive at a new definition for the expert's feature expectations:

$$\hat{\phi}^{Z|Y}_{\boldsymbol{\theta},k} \triangleq \frac{1}{|\mathcal{Y}|} \sum_{Y \in \mathcal{Y}} \sum_{Z \in \mathbb{Z}} P(Z|Y;\boldsymbol{\theta}) \sum_{t=1}^{T} \gamma^t \phi_k(\langle s,a\rangle_t) \tag{2}$$

where $\langle s,a\rangle_t \in Y \cup Z$, $\mathcal{Y}$ is the set of all observed Y, $\mathbb{Z}$ is the set of all possible hidden Z that can complete a trajectory. The program in (1) is modified by replacing $\hat{\phi}_k$ with $\hat{\phi}^{Z|Y}_{\boldsymbol{\theta},k}$, as we show below. Notice that in the case of no occlusion $\mathbb{Z}$ is empty and $\mathcal{X} = \mathcal{Y}$. Therefore $\hat{\phi}^{Z|Y}_{\boldsymbol{\theta},k} = \hat{\phi}_k$ and this method reduces to (1). Thus, this method generalizes the previous maximum entropy IRL method.

$$\begin{array}{l} \max\limits_{\Delta} \left(-\sum_{X \in \mathbb{X}} P(X;\boldsymbol{\theta})\ log\ P(X;\boldsymbol{\theta})\right) \\ \textbf{subject to } \sum_{X \in \mathbb{X}} P(X;\boldsymbol{\theta}) = 1 \\ E_{\mathbb{X}}[\phi_k] = \hat{\phi}^{Z|Y}_{\boldsymbol{\theta},k}\ \forall k \end{array} \tag{3}$$

However, the program in (3) becomes nonconvex due to the presence of $P(Z|Y)$. As such, finding its optima by Lagrangian relaxation is not trivial. Wang et al. [21] suggests a log linear approximation to cast the problem of finding the parameters of distribution (reward weights) as likelihood maximization that can be solved within the

[1] Repeated trajectories in a demonstration can usually be excluded for many methods without impacting the learning.

schema of expectation-maximization [10]. An application of this approach to the problem of IRL under occlusion yields the following two steps (with more details in [7]):

E-step This step involves calculating Eq. 2 to arrive at $\hat{\phi}^{Z|_Y,(t)}_{\boldsymbol{\theta},k}$, a conditional expectation of the K feature functions using the parameter $\boldsymbol{\theta}^{(t)}$ from the previous iteration. We may initialize the parameter vector randomly.

M-step In this step, the modified program (3) is optimized by utilizing $\hat{\phi}^{Z|_Y,(t)}_{\boldsymbol{\theta},k}$ from the E-step above as the expert's constant feature expectations to obtain $\boldsymbol{\theta}^{(t+1)}$. Normalized exponentiated gradient descent [18] solves the program.

As EM may converge to local minima, this process is repeated with random initial $\boldsymbol{\theta}$ and the solution with the maximum entropy is chosen as the final one.

3 INCREMENTAL IRL (I2RL)

We present our framework labeled I2RL in order to realize IRL in an online setting. In addition to presenting previous techniques for online IRL, we introduce a new method that generalizes the maximum entropy IRL under occlusion.

3.1 Framework

To establish the definition of I2RL, we must first define a *session* of I2RL. Let $\hat{R}^0_E$ be an initial estimate of the expert's reward function.

Definition 3 (Session). *A session $\zeta_i(MDP_{/R_E}, \mathcal{X}_i, \hat{R}^{i-1}_E)$, $i > 0$ of I2RL takes as input the expert's MDP sans the reward function, the current (i th) demonstration, $\mathcal{X}_i \in 2^{\mathbb{X}}$, and the reward function estimated previously. It yields a revised estimate of the expert's reward function, $\hat{R}^i_E$.*

Note that we may replace the reward function estimates with some parameter sufficiently representing it (e.g., $\boldsymbol{\theta}$). Also, for expedience in formal analysis, we assume that the trajectories in a session $\mathcal{X}_i$ are i.i.d. from the trajectories in previous session. [2] As the trajectories in $\mathcal{X}_i$ are i.i.d., the demonstrations $\{\mathcal{X}_i, i \in \{1, 2, \ldots\}$ are also i.i.d.

We may let the sessions run indefinitely. Alternately, we may establish some stopping criteria for I2RL, which would allow it to automatically terminate the sessions once the criterion is satisfied. Let $LL(\hat{R}^i_E|\mathcal{X}_{1:i})$ be the log likelihood of the demonstrations received up to the i^{th} session given the current estimate of the expert's reward function. We may view this likelihood as a measure of how well the learned reward function explains the observed data. In the context of I2RL, the log likelihood must be computed without storing data from previous sessions. Here onwards, $\widehat{\mathcal{X}}$ denotes a sufficient statistic that replaces *all input trajectories from previous sessions* in the computation of log likelihood.

Definition 4 (Stopping criterion #1). *Terminate the sessions of I2RL when $|LL(\hat{R}^i_E|\mathcal{X}_i, \widehat{\mathcal{X}}) - LL(\hat{R}^{i-1}_E|\mathcal{X}_{i-1}, \widehat{\mathcal{X}'})| \leqslant \epsilon$, where ϵ is a very small positive number.*

Definition 4 reflects the fact that additional sessions are not improving the learning performance significantly. On the other hand, a more effective stopping criterion is possible if we know the expert's true policy. We utilize the *inverse learning error* [9] in this criterion, which gives the loss of value if learner uses the learned policy on the task instead of the expert's: $ILE(\pi^*_E, \pi_E) = ||V^{\pi^*_E} - V^{\pi_E}||_1$. Here, $V^{\pi^*_E}$ is the optimal value function of E's MDP and V^{π_E} is the value function due to utilizing the learned policy π_E in E's MDP. Notice that when the learned reward function results in an optimal policy identical to E's true policy, $\pi^*_E = \pi_E$, ILE will be zero; it increases monotonically as the two policies increasingly diverge in value. Instead of using an absolute difference, our experiments use a normalized difference, $ILE(\pi^*_E, \pi_E) = \frac{||V^{\pi^*_E} - V^{\pi_E}||_1}{||V^{\pi^*_E}||_1}$. Let π^i_E be the optimal policy obtained from solving the expert's MDP with the reward function $\hat{R}^i_E$ learned in session i (for simpler notation, superscript L is dropped).

Definition 5 (Stopping criterion #2). *Terminate the sessions of I2RL when $ILE(\pi^*_E, \pi^{i-1}_E) - ILE(\pi^*_E, \pi^i_E) \leqslant \epsilon$, where ϵ is a very small positive error and is given.*

Obviously, prior knowledge of the expert's policy is not common. Therefore, we view this criterion as being more useful during the formative assessments of I2RL methods. Utilizing Defs. 3, 4, and 5, we formally define I2RL next.

Definition 6 (I2RL). *Incremental IRL is a sequence of learning sessions $\{\zeta_1(MDP_{/R_E}, \mathcal{X}_1, \hat{R}^0_E), \zeta_2(MDP_{/R_E}, \mathcal{X}_2, \hat{R}^1_E), \zeta_3\,(MDP_{/R_E}, \mathcal{X}_3, \hat{R}^2_E), \ldots, \}$, which continue infinitely, or until a stopping criterion assessing convergence is met (criterion #1 or #2 depending on which one is chosen a'priori).*

While somewhat straightforward, these rigorous definitions for I2RL allow us to situate the few existing online IRL techniques, and to introduce online IRL with hidden variables, as we see next.

3.2 Methods

One of our contributions is to facilitate a portfolio of online methods each with its own appealing properties under the framework of I2RL. This will enable online IRL in various applications. An early method for online IRL [12] modifies Ng and Russell's linear program [13] to take as input a single trajectory (instead of a policy) and replaces the linear program with an incremental update of the reward function. We may easily present this method within the framework of I2RL. A session of this method $\zeta_i(MDP_{/R_E}, \mathcal{X}_i, \hat{R}^{i-1}_E)$ is realized as follows: Each $\mathcal{X}_i$ is a single state-action pair $\langle s, a\rangle$ and initial reward function $\hat{R}^0_E = \frac{1}{\sqrt{|S_E|}}$. For $i > 0$, $\hat{R}^i_E = \hat{R}^{i-1}_E + \alpha \cdot v_i$, where v_i is the difference in expected value of the observed action a at state s and the (predicted) optimal action obtained by solving the MDP with the reward function $\hat{R}^{i-1}_E$, and α is the learning rate. While no explicit stopping criterion is specified, the incremental method terminates when it runs out of observed state-action pairs. Jin et al. [12] provide the algorithm for this method as well as error bounds.

A recent method by Rhinehart et al. [16] performs online IRL for activity forecasting. Casting this method to the framework of I2RL, a session of this method is $\zeta_i(MDP_{/R_E}, \mathcal{X}_i, \boldsymbol{\theta}^{i-1})$, which yields $\boldsymbol{\theta}^i$. Input demonstration for the session, $\mathcal{X}_i$, comprises all the activity trajectories observed since the end of previous goal until the next goal is reached. The session IRL finds the reward weights $\boldsymbol{\theta}^i$ that

[2] The assumption holds when each session starts from the same state. In case of occlusion, even though inferring the hidden portion Z of a trajectory $X \in \mathcal{X}_i$, is influenced by the visible portion, Y, this does not make the trajectories necessarily dependent on each other.

minimize the margin $\phi^{\pi^*_E} - \hat{\phi}$ using gradient descent. Here, the expert's policy π^*_E is obtained by using soft value iteration for solving the complete MDP that includes a reward function estimate obtained using previous weights $\boldsymbol{\theta}^{i-1}$. No stopping criterion is utilized for the online learning, thereby emphasizing its continuity.

3.2.1 Incremental Latent MaxEnt. We present a new method for online IRL under the I2RL framework, which modifies the latent maximum entropy (LME) optimization reviewed in the Background section. It offers the capability to perform online IRL in contexts where portions of the observed trajectory may be occluded.

For differentiation, we refer to the original method as the *batch* version. Recall the k^{th} feature expectation of the expert computed in Eq. 2 as part of the E-step. $\hat{\phi}^{Z|_Y,i}_{\boldsymbol{\theta}^i,k}$ is the expectation of k^{th} feature for the demonstration obtained in i^{th} session, $\hat{\phi}^{Z|_Y,1:i}_{\boldsymbol{\theta}^i,k}$ is the expectation computed for all demonstrations obtained till ith session, we may rewrite Eq. 2 for feature k as:

$$
\begin{aligned}
\hat{\phi}^{Z|_Y,1:i}_{\boldsymbol{\theta}^i,k} &\triangleq \frac{1}{|\mathcal{Y}_{1:i}|} \sum_{Y\in\mathcal{Y}_{1:i}} \sum_{Z\in\mathbb{Z}} P(Z|Y;\boldsymbol{\theta}) \sum_{t=1}^{T} \gamma^t \phi_k(\langle s,a\rangle_t) \\
&= \frac{1}{|\mathcal{Y}_{1:i}|} \Bigg(\sum_{Y\in\mathcal{Y}_{1:i-1}} \sum_{Z\in\mathbb{Z}} P(Z|Y;\boldsymbol{\theta}) \sum_{t=1}^{T} \gamma^t \phi_k(\langle s,a\rangle_t) + \\
&\sum_{Y\in\mathcal{Y}_i} \sum_{Z\in\mathbb{Z}} P(Z|Y;\boldsymbol{\theta}^i) \sum_{t=1}^{T} \gamma^t \phi_k(\langle s,a\rangle_t) \Bigg) \\
&= \frac{1}{|\mathcal{Y}_{1:i-1}| + |\mathcal{Y}_i|} \left(|\mathcal{Y}_{1:i-1}|\, \hat{\phi}^{Z|_Y,1:i-1}_{\boldsymbol{\theta}^{i-1},k} + |\mathcal{Y}_i|\, \hat{\phi}^{Z|_Y,i}_{\boldsymbol{\theta}^i,k} \right) \\
&\text{(Using Eq. 2 and } |\mathcal{Y}_{1:i}| = |\mathcal{Y}_{1:i-1}| + |\mathcal{Y}_i|\text{)} \qquad (4)
\end{aligned}
$$

A session of incremental LME takes as input the expert's MDP sans the reward function, the current session's trajectories, the number of trajectories observed until previous session, the expert's empirical feature expectation and reward weights from previous session. More concisely, each session is denoted by, $\zeta_i(MDP_{/R_E}, \mathcal{Y}_i, |\mathcal{Y}_{1:i-1}|, \hat{\phi}^{Z|_Y,1:i-1}_{\boldsymbol{\theta}^{i-1}}, \boldsymbol{\theta}^{i-1})$. The sufficient statistic $\widehat{X}$ for the session comprises $(|\mathcal{Y}_{1:i-1}|, \hat{\phi}^{Z|_Y,1:i-1}_{\boldsymbol{\theta}^{i-1}})$. In each session, the feature expectations using that session's observed trajectories are computed, and the output feature expectations are obtained by including these as shown above in Eq. 4; the latter is used in the M-step. The equation shows how computing sufficient statistic replaces the need for storing the data input in previous sessions. Of course, each session may involve several iterations of the E- and M-steps until the converged reward weights $\boldsymbol{\theta}^i$ are obtained thereby giving the corresponding reward function estimate. We refer to this method as LME I2RL.

Wang et al. [20] shows that if the distribution over the trajectories in (3) is log linear, then the reward function that maximizes the entropy of the trajectory distribution also maximizes the log likelihood of the observed portions of the trajectories. Given this linkage with log likelihood, the stopping criterion #1 as given in Def. 4 can be utilized. As shown in Algorithm 1, the sessions will terminate when, $|LL(\boldsymbol{\theta}^i|\mathcal{Y}_i, |\mathcal{Y}_{1:i-1}|, \hat{\phi}^{Z|_Y,1:i-1}_{\boldsymbol{\theta}^{i-1}}, \boldsymbol{\theta}^{i-1}) - LL(\boldsymbol{\theta}^{i-1}|\mathcal{Y}_{i-1}, |\mathcal{Y}_{1:i-2}|, \hat{\phi}^{Z|_Y,1:i-2}_{\boldsymbol{\theta}^{i-2}}, \boldsymbol{\theta}^{i-2})| \leq \epsilon$, where $\boldsymbol{\theta}^i$ fully parameterizes the reward function estimate for the i^{th} session and ϵ is a given acceptable difference.

Algorithm 1 Algorithm INCREMENTAL-LME($MDP_{/R_E}$,ϕ)

```
i ← 1; $\mathcal{Y}_{1:i-1} \leftarrow \emptyset$
$\hat{\phi}^{Z|_Y,1:i-1}_{\boldsymbol{\theta}^{i-1},k} \leftarrow 0$; $[\boldsymbol{\theta}^0]_k \sim$ uniform(0, 1)
while $|LL(\mathcal{Y}_i, |\mathcal{Y}_{i-1}|, \hat{\phi}^{Z|_Y,1:i-1}_{\boldsymbol{\theta}^{i-1}}, \boldsymbol{\theta}^i) - LL(\mathcal{Y}_{i-1}, |\mathcal{Y}_{i-2}|, \hat{\phi}^{Z|_Y,1:i-2}_{\boldsymbol{\theta}^{i-2}}, \boldsymbol{\theta}^{i-1})| \leq \varepsilon$ do
  /* session $\zeta_i(M_{/R_E}, \mathcal{Y}_i, |\mathcal{Y}_{1:i-1}|, \hat{\phi}^{Z|_Y,1:i-1}_{\boldsymbol{\theta}^{i-1}}, \boldsymbol{\theta}^{i-1})$ */
  repeat
    /* E-step */
    Use MCMC to sample trajectories from $P((Y,Z)|\boldsymbol{\theta}^{i-1})$, and compute $\hat{\phi}^{Z|_Y,i}_{\boldsymbol{\theta}^i}$ for sampled trajectories.
    /* Updating feature expectations using sufficient statistic. */
    Use Equation 4 to compute $\hat{\phi}^{Z|_Y,1:i}_{\boldsymbol{\theta}^i,k}$ for all $k$.
    $|\mathcal{Y}_{1:i}| \leftarrow |\mathcal{Y}_{1:i-1}| + |\mathcal{Y}_i|$
    /* M-step */
    $\theta_0 \leftarrow \boldsymbol{\theta}^{i-1}, t \leftarrow 1$
    repeat
      Compute $\pi^*_{E,(t-1)}$ using $\boldsymbol{\theta}_{(t-1)}$ and $E_{\mathbb{X}}[\phi_k]$ using trajectories sampled from $\pi^*_{E,(t-1)}$.
      $z_{(t-1)} \leftarrow \hat{\phi}^{Z|_Y,1:i}_{\boldsymbol{\theta}^i} - E_{\mathbb{X}}[\phi]$   {gradient}
      $\boldsymbol{\theta}_{t,k} \leftarrow \frac{\boldsymbol{\theta}_{(t-1),k} \exp(-\eta z_{(t-1),k})}{\sum_{i=1}^{k} \boldsymbol{\theta}_{(t-1),k} \exp(-\eta z_{(t-1),k})}$
      $t \leftarrow t + 1$
    until $|\boldsymbol{\theta}_t| \approx |\boldsymbol{\theta}_{t-1}|$
  until gradient of likelihood ≈ 0
  Compute $\hat{\pi}_i$ using learned reward $\boldsymbol{\theta}^i \leftarrow \boldsymbol{\theta}_t$.
  $i \leftarrow i + 1$
```

3.3 Convergence Bounds

LME I2RL admits some significant convergence guarantees with a confidence of meeting the specified error on the demonstration likelihood. We defer the proofs of these results to the supplementary file available at https://tinyurl.com/yyywmx9x. To establish the guarantees of LME I2RL, we first focus on the full observability setting. For a desired bound ε on the log-likelihood loss (difference in likelihood w.r.t. expert's true $\boldsymbol{\theta}_E$ and that w.r.t learned $\boldsymbol{\theta}^i$) for session i, the confidence is bounded as follows:

Theorem 1 (Confidence for ME I2RL). *Given $\mathcal{X}_{1:i}$ as the (fully observed) demonstration till session i, $\boldsymbol{\theta}_E \in [0,1]^K$ is the expert's weights, and $\boldsymbol{\theta}^i$ is the converged weight vector for session i for ME I2RL, we have,*

$$
LL(\boldsymbol{\theta}_E|\mathcal{X}_{1:i}) - LL(\boldsymbol{\theta}^i|\mathcal{X}_i, |\mathcal{X}_{i-1}|, \hat{\phi}^{1:i-1}, \boldsymbol{\theta}^{i-1}) \leqslant \frac{2K\varepsilon}{(1-\gamma)}
$$

with probability atleast $\max(0, 1-\delta)$, *where* $\delta = 2K\exp(-2|\mathcal{X}_{1:i}|\varepsilon^2)$.

Note that sufficient statistic $\hat{X}$ for full-observability scenario is $(|\mathcal{X}_{i-1}|, \hat{\phi}^{1:i-1})$. Theorem 1 holds for the online method by Rhinehart et al. [16] because it uses incremental (full-observability) maximum entropy IRL. As the latter implements online learning without an incremental update of feature expectations of the expert, thus set $\hat{\phi}^{1:i} = \hat{\phi}^{i}$, an absence of sufficient statistic, set $|\mathcal{X}_{i-1}| = 0$, and set $\hat{\phi}_k^{1:i-1} = 0, \forall k$ in Theorem 1. This demonstrates the benefit of Theorem 1 to relevant methods.

Relaxing the full observability assumption, the following lemma proves that LME I2RL converges monotonically.

Lemma 1 (Monotonicity). *LME I2RL increases the demonstration likelihood monotonically with each new session,* $LL(\boldsymbol{\theta}^i | \mathcal{Y}_i, |\mathcal{Y}_{1:i-1}|, \hat{\phi}_{\boldsymbol{\theta}^{i-1}}^{Z|_Y,1:i-1}, \boldsymbol{\theta}^{i-1}) - LL(\boldsymbol{\theta}^{i-1} | \mathcal{Y}_{i-1}, |\mathcal{Y}_{1:i-2}|, \hat{\phi}_{\boldsymbol{\theta}^{i-2}}^{Z|_Y,1:i-2}, \boldsymbol{\theta}^{i-2}) \geqslant 0$, *when* $|\mathcal{Y}_{1:i-1}| \gg |\mathcal{Y}_i|$.

While Lemma 1 suggests that the log likelihood of the demonstration can only improve from session to session after learner has accumulated a significant amount of observations, a stronger result illuminates the confidence with which LME I2RL approaches, over a sequence of sessions, the log likelihood of the expert's true weights $\boldsymbol{\theta}_E$. As a step toward such a result, we first consider the error in approximating the feature expectations of the unobserved portions of the data, accumulated from the first to the current session of I2RL. Notice that $\hat{\phi}_{\boldsymbol{\theta}^i,k}^{Z|_Y,1:i}$ given by Eq. 4 is an approximation of the full-observability expectation $\hat{\phi}_k^{1:i}$, computed by sampling the hidden Z from $P(Z|Y, \boldsymbol{\theta}^{i-1})$ [7]. The following lemma relates the error due to this sampling-based approximation, i.e., $\left|\hat{\phi}_k^{1:i} - \hat{\phi}_{\boldsymbol{\theta}^i,k}^{Z|_Y,1:i}\right|$, to the difference between feature expectations for learned policy and that estimated for the expert's true policy.

Lemma 2 (Constraint Bounds for LME I2RL). *Suppose* $\mathcal{X}_{1:i}$ *has portions of trajectories in* $\mathbb{Z}_{1:i} = \{Z|(Y,Z) \in \mathcal{X}_{1:i}\}$ *occluded from the learner. Let* ε_s *be a bound on the error* $\left|\hat{\phi}_k^{1:i} - \hat{\phi}_{\boldsymbol{\theta}^i,k}^{Z|_Y,1:i}\right|_1, k \in \{1, 2 \ldots K\}$ *after* n_s *samples for approximation. Then, with probability at least* $\max(0, 1 - (\delta + \delta_s))$, *the following holds:*

$$\left|E_{\mathbb{X}}[\phi_k] - \hat{\phi}_{\boldsymbol{\theta}^i,k}^{Z|_Y,1:i}\right|_1 \leqslant \varepsilon + \varepsilon_s, k \in \{1, 2 \ldots K\}$$

where ε, δ *are as defined in Theorem 1, and* $\delta_s = 2K \exp(-2n_s\varepsilon_s^2)$.

LME I2RL computes θ^i by an optimization process using the result $\phi^{Z|_Y,i}$ of E step (sampling of occluded data) of current session along with other inputs (feature expectations and θ computed from previous session) which, in turn, depend on sampling process in previous sessions. Theorem 1 and Lemma 2 allows us to probabilistically bound the error in log likelihood for LME I2RL:

Theorem 2 (Confidence for LME I2RL). *Let* $\mathcal{Y}_{1:i} = \{Y|(Y,Z) \in \mathcal{X}_{1:i}\}$ *be the observed portions of the demonstration until session i.* ε *and* ε_s *are inputs as defined in Lemma 2, and* $\boldsymbol{\theta}^i$ *is the solution of session i for LME I2RL. Then*

$$LL(\boldsymbol{\theta}_E | \mathcal{Y}_{1:i}) - LL(\boldsymbol{\theta}^i | \mathcal{Y}_i, |\mathcal{Y}_{i-1}|, \hat{\phi}_{\boldsymbol{\theta}^{i-1}}^{Z|_Y,1:i-1}, \boldsymbol{\theta}^{i-1}) \leq \frac{4K\varepsilon_l}{(1-\gamma)}$$

with confidence at least $\max(0, 1 - \delta_l)$, *where* $\varepsilon_l = \frac{\varepsilon + \varepsilon_s}{2}$, *and* $\delta_l = \delta + \delta_s$.

Given $\varepsilon, \varepsilon_s, N$ and the total number of input partial -trajectories, $|\mathcal{Y}_{1:i}|$, Theorem 2 gives the confidence $1 - \delta_l$ for I2RL under occlusion. Equivalently, $|\mathcal{Y}_{1:i}|$ can be derived using desired error bounds and confidence. As a boundary case of LME I2RL, if learner ignores occluded data (no sampling or $n_s = 0$ for E-step), the confidence for convergence becomes zero because δ_s becomes larger than 1.

4 EXPERIMENTS

We evaluate the benefit of online IRL on the perimeter patrol domain, introduced by Bogert and Doshi [4] for evaluating IRL, and simulated in ROS Player Stage using data and files made publicly available. It involves a robotic learner observing two patrollers continuously patrol a hallway as shown in Fig. 1 (left). The learner is tasked with reaching the cell marked 'G' (Fig. 1 right) without being spotted by any of the patrollers. Each guard can see up to 3 grid cells in front. This domain differs from the usual applications of IRL toward imitation learning. In particular, the learner must solve its own distinct decision-making problem (modeled as another MDP) that is reliant on knowing how the guards patrol, which can be estimated from inferring each guard's preferences. The grid is broadly divided into 5 regions and guard MDPs utilized two types of binary state-action features: does the current action in the current state make the guard change its grid cell?, is robot turning around in cell (x, y) in a given region of grid? One movement based feature and 5 turning around features leads to a total of six. The true weight vector $\boldsymbol{\theta}_E$ for these features is $\langle .57, 0, 0, 0, .43, 0 \rangle$. These weights assign the highest preference to actions that constantly change the grid cell, and the next preference to turning in smaller upper and lower hallways (Fig. 1left), which leads to a reward function that makes two guards move back and forth constantly.

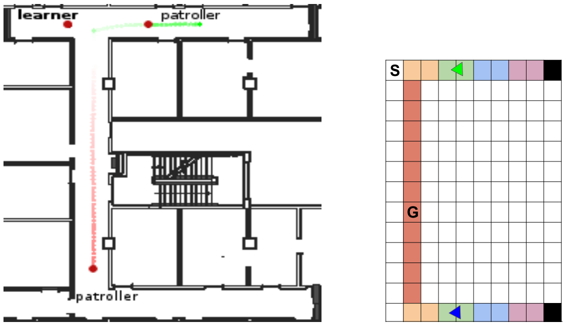
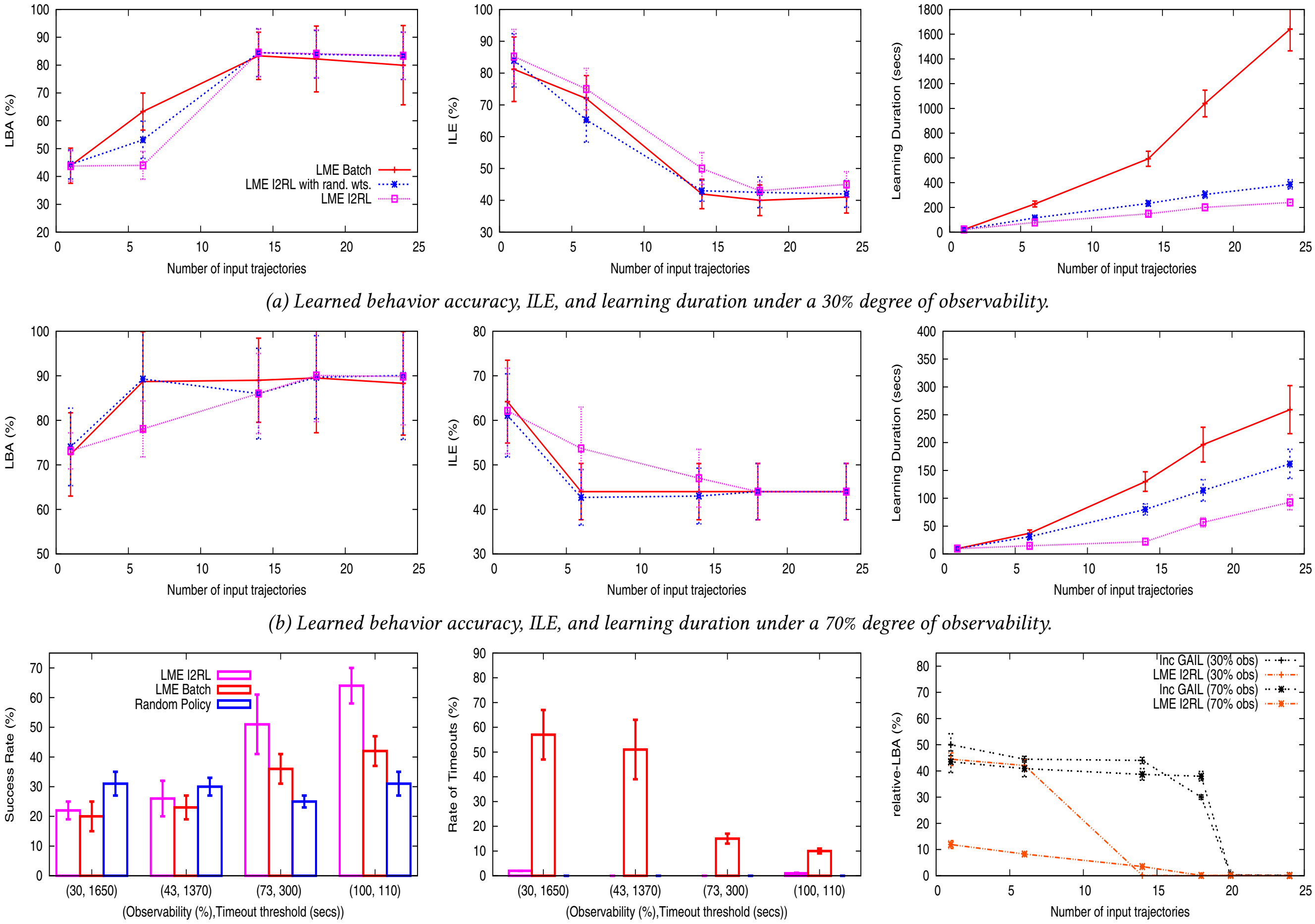


Figure 1: The map and the corresponding MDP state space for each patroller [4]. MDP has 3 dimensional state space (x,y,orientation) with 124 states, and it has 4 actions (move-forward, turn left, turn right, stop). The color-shaded regions (long hallway, turning points and 3 small divisions in small hallways on both sides) are the 5 regions defining movement and turn-around features. S and G are start and goal locations for the learner. Simulations were run on a Ubuntu 14 LTS system with an Intel i5 2.8GHz CPU core and 8GB RAM. Learner is unaware of where each patroller turns around or their navigation capabilities.

As the learner's vantage point limits its observability, this domain requires IRL under occlusion. To establish the benefit of I2RL, we compare the performances of both *batch* and *incremental* variants of LME method. These methods are applicable to both finite- and infinite-horizon MDPs when we interpret horizon as the look ahead.

Theorem 2 allows us to derive an upper bound on the size of the input needed across all sessions to meet the given log likelihood error, which signals convergence. Table 1 (a) shows this relation



(a) Learned behavior accuracy, ILE, and learning duration under a 30% degree of observability.

(b) Learned behavior accuracy, ILE, and learning duration under a 70% degree of observability.

(c) Success rates and timeouts under both 30%, 70%, and full observability. The success rate obtained by a random baseline is shown as well. Method random-policy does not perform IRL and picks a random set of reward weights for computing expert's policy. Rightmost chart shows the relative difference computed as (LBA for full observability - LBA under occlusion)/(LBA for full observability) for both 30% and 70% observability.

Figure 2: Various metrics for comparing the performances of batch and incremental LME on Bogert and Doshi's [4] perimeter patrolling domain.

ϵ_l (ϵ, ϵ_s)	$\lvert\mathcal{Y}_{1:i}\rvert$
0.125 (0.2, 0.05)	60
0.075 (0.1, 0.05)	239
0.05 (0.05, 0.05)	957
0.045 (0.04, 0.05)	1496

(a)

$\lvert\mathcal{Y}_{1:i}\rvert$	max $(0, 1 - \delta_l)$
115	0
135	0.19
200	0.78
375	0.99

(b)

Table 1: (a) Number of trajectories required for ϵ_l convergence in the patrolling domain ($K = 6, \gamma = 0.99$) with confidence $1 - \delta_l = 1 - (\delta + \delta_s) = 1 - (0.1 + 0.1) = 0.8$. We use $\epsilon_s = 0.05$ for both 30% and 70% observability. (b) Confidence of convergence increases with more trajectories (from more sessions) with $\epsilon_l = 0.075$.

between the acceptable error ϵ_l, which is a function of ϵ and ϵ_s, and the number of trajectories for 80% confidence. Furthermore, the maximum number of MCMC samples required in each E-step are $N = -\frac{1}{2\epsilon_s^2} \ln \frac{\delta_s}{2K} = 957$. We pick $\epsilon_l = 0.075$ for our experiments, and Table 1(a) shows that at most 239 trajectories would be required. Table 1(b) shows that, for chosen value of ϵ_l, the confidence of convergence increases with more sessions.

Efficacy of the methods was compared using the following metrics: *learned behavior accuracy* (*LBA*), which is the proportion of all states at which the actions prescribed by the inversely learned policies of both patrollers coincide with their actual actions; *ILE*, which was defined previously; and *success rate*, which is the percentage of runs where L reaches the goal state undetected. Note that when the learned behavior accuracy is high, we expect the ILE to be low. However, as MDPs admit multiple optimal policies, a low

ILE need not translate into a high behavior accuracy. As such, these two metrics are not strictly correlated.

We report the LBA, ILE, and the computation time for learning process (learning duration in seconds) of the inverse learning for both batch and incremental LME in Figs. $2(a)$ and $2(b)$; the former under a 30% degree of observability and the latter under 70%. For a fair comparison, we give exactly same data as input to both methods. Each data point is averaged over 100 trials for a fixed degree of observability and a fixed number of trajectories in the demonstration $\mathcal{X}$. While the entire demonstration is given as input to the batch variant, the $\mathcal{X}_i$ for each session of I2RL has one trajectory composed of 5 state-action pairs. As such, the incremental learning stops when there are no more trajectories remaining to be processed. To better understand any differentiations in performance, we introduce a third variant that implements each session as, $\zeta_i(MDP_{/R_E}, \mathcal{Y}_i, |\mathcal{Y}_{i:i-1}|, \hat{\phi}_{\theta^i}^{Z|_Y,1:i-1})$. Notice that this incremental variant does not utilize the previous session's reward weights, instead it initializes them randomly in each session; we label it as LME I2RL *with random weights*.

We empirically verify that convergence is indeed achieved within 239 sessions (each having one trajectory). As the size of demonstration increases, both batch and incremental variants exhibit similar quality of learning although initially the incremental performs slightly worse. Importantly, LME I2RL achieves these learning accuracies in significantly less time compared to batch, with the speed up ratio increasing to four as $|\mathcal{X}|$ grows. On the other hand, the batch method generally fails to converge in the total time taken by the incremental variant. Notice that a random initialization of weights in each session, performed in LME I2RL with random weights, leads to higher learning durations as we may expect. A video of a simulation run of the multi-robot patrolling domain is available at https://youtu.be/B3wA6z111ws.

Is there a benefit due to the reduced learning time? We show the success rates of the learner when each of the three methods are utilized for IRL in Fig. $2(c)$. LME I2RL begins to demonstrate comparatively better success rates under 30% observability itself, which further improves when the observability is at 70%. While the batch LME's success rate does not exceed 40%, the incremental variant succeeds in reaching the goal location undetected in about 65% of the runs under full observability (the last data point). A deeper analysis in order to understand these differences in success rates between batch and incremental generalization of LME reveals that batch LME suffers from a large percentage of *timeouts* – more than 50% for low observability, which drops down to 10% for full observability. A timeout occurs when IRL fails to converge to a reward estimate in a reasonable amount of time for each run. We compute the threshold for timeout as the total time taken for perception of trajectories, learning, and two rounds of patrolling averaged over many trials, which gives both Batch IRL and I2RL at least two chances for penetrating the patrol. LME with low observability requires more time due to the larger portion of the trajectory being hidden, which requires sampling a larger trajectory for computing expectation. On the other hand, incremental LME suffers from very few timeouts. Of course, other factors play secondary roles in success as well.

We compare the performance of LME I2RL with that of an online version of GAIL [11], a state-of-the-art policy learning method cast in the schema of generative adversarial networks. We experimented with various simulation settings eventually settling on one that seemed most appropriate for our domain (500 iterations of TRPO with an adversary-batch-size of 1,000, 2 hidden-layer $[64 \times 8]$ network for both generator and adversary, adversary-epochs = 5, and generator-batch-size = 150). We obtained a maximum LBA of 52% for the fully observable simulations (note that fully observable trajectories still may not yield all state-action pairs). This absolute performance being rather low, we analyzed the relative impact of occlusion in our scenario on the performance of GAIL. Figure 2 (c) shows that while both LME I2RL and online GAIL demonstrate the same relative difference initially, the latter method requires significantly more trajectories before it catches up with its full-observability performance, for both the 30% and 70% observability cases. As such, online GAIL appears to be far more impacted by occlusion that LME I2RL.

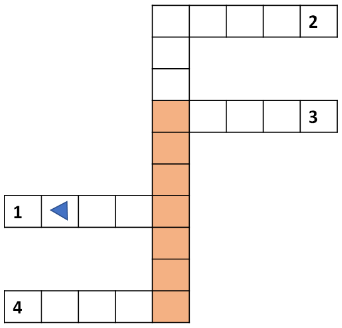


Figure 3: A single patroller denoted by the triangle moves clockwise to the ends of four hallways in the numbered order, and just the shaded area is visible to the learner.

In order to evaluate the scalability of LME I2RL, we compare the learning durations of batch and incremental methods in a larger patrolling domain with 192 states. Previous experiments establish that the success rate is primarily predicated on the learning performance. Therefore, we focus on related metrics. For this domain, Player Stage simulator has not been used. Instead, we utilize a set of trajectories obtained by sampling expert's policy directly. The grid is divided into 4 regions corresponding to the ends of four hallways. Patrollers' reward function utilized four features, each activating when it switches target from end of one hallway to next one in a clockwise fashion (Fig. 3). The MDP's state includes information of current location and last visited region. Equal weights are given to each feature, which makes the patroller move through grid clockwise to activate them. The learner perceives just 32% of total states. As shown in Fig. 4, LME I2RL achieves the same accuracy – measured by LBA and ILE – as batch LME but in significantly less time.

How well do these results extend to physical robots? We conducted the perimeter patrol experiment on physical Turtlebots in the actual hallway shown in Fig. 1 to verify the benefits of I2RL in a real-world setting (Fig 5). The learner observes less than 30% of the patrols.

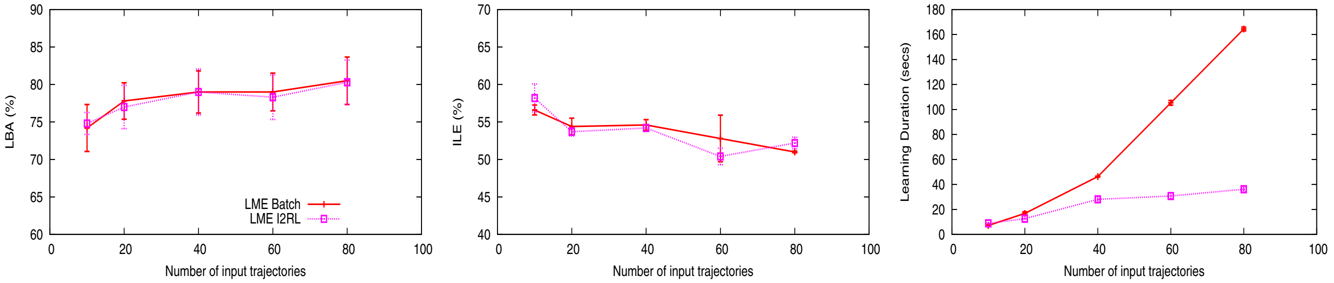


Figure 4: Performances of batch and incremental LME on various metrics for the larger domain.

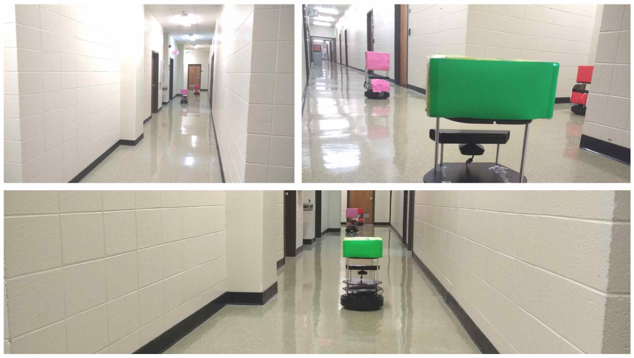


Figure 5: In counterclockwise direction: patrollers (in pink and red) in the longer hallway. Learner (green) observing them from its vantage point in the smaller hallway, and learner breaching patrol to reach the goal (first door to its right).

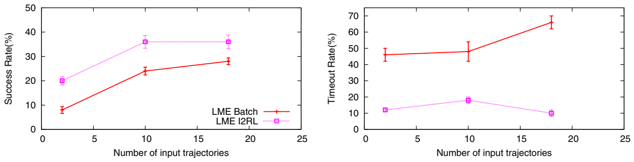


Figure 6: Success and timeout rates for experiments involving physical robots at less than 30% observability.

The states of patrollers were recognized via blob detection using the CMVision ROS package. The threshold for timeout is set the same as that in simulations. Though degree of observability cannot be changed here, we vary the number of input trajectories to observe the change in success and timeout rates. Figure 2 gives a comparison between LME in batch and online versions for various metrics with each data point averaged across 5 sets of 10 trials each. While the overall success rate is not high, LME I2RL continues to penetrate more patrols successfully than batch and exhibits a much reduced time out rate.

5 CONCLUDING REMARKS

This paper contributes to the nascent problem of online IRL by offering a formal framework, I2RL, to help analyze the class of methods for online IRL. I2RL facilitates comparing various online IRL techniques and facilitates establishing the theoretical properties of online methods. In particular, it provides a common ground for researchers interested in developing techniques for online IRL.

We presented a new method within the I2RL framework that generalizes recent advances in maximum entropy IRL to online settings. Casting this method to the context of I2RL allowed us to establish key theoretical properties of (full-observability) maximum entropy I2RL and LME I2RL, ensuring the desired monotonic progress with a given confidence of convergence. Lemma 2 utilizes user-specified ε and ε_s to bound the key gradient $(\hat{\phi} - E_X[\phi])$ used in the likelihood maximization process, and Theorem 2 bounds the error in log likelihood of the reward parameters due to incremental learning. As batch IRL can be viewed as a specific case of I2RL having just one session, the theoretical results trivially hold for batch LME as well.

Our comprehensive experiments show that the new I2RL method improves over the previous state-of-the-art batch method in time-limited domains, by approximately reproducing the batch method's accuracy but in significantly less time. In particular, we have shown that given the practical constraints on computation time exhibited by an online IRL application, the new method is able to solve the problem with a higher success rate. This IRL generalization also suffers less from occlusion than methods that directly learn the policy or behavior. Future avenues for investigation include understanding how I2RL can address some of the challenges related to player Stage simulation of larger domain, as well as I2RL without prior knowledge of dynamics of the experts.

6 ACKNOWLEDGMENTS

We thank Kenneth Bogert for insightful discussions, and the anonymous reviewers for helpful comments and suggestions. This work was supported in part by a research contract with the Toyota Research Institute of North America (TRI-NA), and by National Science Foundation grants IIS-1830421 and IIS-1526813.

REFERENCES

[1] Pieter Abbeel and Andrew Y. Ng. 2004. Apprenticeship Learning via Inverse Reinforcement Learning. In *Twenty-first International Conference on Machine Learning (ICML)*. 1–8.

[2] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. 2009. A survey of robot learning from demonstration. *Robotics and autonomous systems* 57, 5 (2009), 469–483.

[3] Monica Babes-Vroman, Vukosi Marivate, Kaushik Subramanian, and Michael Littman. 2011. Apprenticeship learning about multiple intentions. In *28th International Conference on Machine Learning (ICML)*. 897–904.

[4] Kenneth Bogert and Prashant Doshi. 2014. Multi-robot Inverse Reinforcement Learning Under Occlusion with Interactions. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems (AAMAS '14)*. 173–180.

[5] Kenneth Bogert and Prashant Doshi. 2015. Toward Estimating Others' Transition Models Under Occlusion for Multi-robot IRL. In *24th International Joint Conference on Artificial Intelligence (IJCAI)*. 1867–1873.

[6] Kenneth Bogert and Prashant Doshi. 2017. Scaling Expectation-Maximization for Inverse Reinforcement Learning to Multiple Robots Under Occlusion. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems (AAMAS '17)*. 522–529.

[7] Kenneth Bogert, Jonathan Feng-Shun Lin, Prashant Doshi, and Dana Kulic. 2016. Expectation-Maximization for Inverse Reinforcement Learning with Hidden Data. In *2016 International Conference on Autonomous Agents and Multiagent Systems*. 1034–1042.

[8] Abdeslam Boularias, Oliver Krömer, and Jan Peters. 2012. Structured Apprenticeship Learning. In *European Conference on Machine Learning and Knowledge Discovery in Databases, Part II*. 227–242.

[9] Jaedeug Choi and Kee-Eung Kim. 2011. Inverse Reinforcement Learning in Partially Observable Environments. *J. Mach. Learn. Res.* 12 (2011), 691–730.

[10] A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)* 39 (1977), 1–38. Issue 1.

[11] Jonathan Ho and Stefano Ermon. 2016. Generative Adversarial Imitation Learning. In *Advances in Neural Information Processing Systems (NIPS) 29*. 4565–4573.

[12] Zhuo jun Jin, Hui Qian, Shen yi Chen, and Miao liang Zhu. 2010. Convergence Analysis of an Incremental Approach to Online Inverse Reinforcement Learning. *Journal of Zhejiang University - Science C* 12, 1 (2010), 17–24.

[13] Andrew Ng and Stuart Russell. 2000. Algorithms for inverse reinforcement learning. In *Seventeenth International Conference on Machine Learning*. 663–670.

[14] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J. Andrew Bagnell, Pieter Abbeel, and Jan Peters. 2018. An Algorithmic Perspective on Imitation Learning. *Foundations and Trends® in Robotics* 7, 1-2 (2018), 1–179.

[15] Deepak Ramachandran and Eyal Amir. 2007. Bayesian Inverse Reinforcement Learning. In *20th International Joint Conference on Artifical Intelligence (IJCAI)*. 2586–2591.

[16] Nicholas Rhinehart and Kris M. Kitani. 2017. First-Person Activity Forecasting with Online Inverse Reinforcement Learning. In *International Conference on Computer Vision (ICCV)*.

[17] Stuart Russell. 1998. Learning Agents for Uncertain Environments (Extended Abstract). In *Eleventh Annual Conference on Computational Learning Theory*. 101–103.

[18] Jacob Steinhardt and Percy Liang. 2014. Adaptivity and Optimism: An Improved Exponentiated Gradient Algorithm. In *31st International Conference on Machine Learning*. 1593–1601.

[19] M. Trivedi and P. Doshi. 2018. Inverse Learning of Robot Behavior for Collaborative Planning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 1–9.

[20] Shaojun Wang, Ronald Rosenfeld, Yunxin Zhao, and Dale Schuurmans. 2002. The Latent Maximum Entropy Principle. In *IEEE International Symposium on Information Theory*. 131–131.

[21] Shaojun Wang and Dale Schuurmans Yunxin Zhao. 2012. The Latent Maximum Entropy Principle. *ACM Transactions on Knowledge Discovery from Data* 6, 8 (2012).

[22] Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey. 2008. Maximum Entropy Inverse Reinforcement Learning. In *23rd National Conference on Artificial Intelligence - Volume 3*. 1433–1438.