Approximate MaxEnt Inverse Optimal Control and its Application for Mental Simulation of Human Interactions

De-An Huang and **Amir-massoud Farahmand** and **Kris M. Kitani** and **J. Andrew Bagnell**
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Maximum entropy inverse optimal control (MaxEnt IOC) is an effective means of discovering the underlying cost function of demonstrated human activity and can be used to predict human behavior over low-dimensional state spaces (i.e., forecasting of 2D trajectories). To enable inference in very large state spaces, we introduce an approximate MaxEnt IOC procedure to address the fundamental computational bottleneck stemming from calculating the partition function via dynamic programming. Approximate MaxEnt IOC is based on two components: approximate dynamic programming and Monte Carlo sampling. We analyze this approximation approach and provide a finite-sample error upper bound on its excess loss. We validate the proposed method in the context of analyzing dual-agent interactions from video, where we use approximate MaxEnt IOC to simulate mental images of a single agents body pose sequence (a high-dimensional image space). We experiment with sequences image data taken from RGB and RGBD data and show that it is possible to learn cost functions that lead to accurate predictions in high-dimensional problems that were previously intractable.

Introduction

The Maximum Entropy (MaxEnt) Inverse Optimal Control (IOC) framework is an effective approach for discovering the underlying reward model of a rational agent and enables robust sequence prediction over low-dimensional state spaces (Ziebart et al. 2008; Ziebart, Bagnell, and Dey 2013). The IOC framework is particularly useful in the context of understanding and modeling human activities, where the recovered reward model intuitively encodes a person's set of preferences. Furthermore, in the MaxEnt formulation of IOC, the soft-maximum value function (log-partition function) compactly describes a global distribution over every possible action sequence. The log-partition function can then be used to simulate and forecast human activities.

Of particular interest in this paper is recent work fusing computer vision and IOC to mentally (visually) simulate human activities. By integrating visual attributes of the scene as features of the reward function, it was shown that highly accurate pedestrian trajectories can be simulated in novel scenes (Kitani et al. 2012). The application of IOC to visual prediction problems, however, has been limited to 2D pedestrian trajectories since current approaches only work for problems with small state space. To extend IOC to deal with the inherent high-dimensional nature of observed human activity from image data, previous approaches (Huang and Kitani 2014; Walker, Gupta, and Hebert 2014) relied on clustering techniques to quantize and reduce the size of the state space. However, coarse discretization of the state space resulted in non-smooth trajectories and inhibited the model's power to simulate the subtle qualities of activity dynamics.

At the heart of the problem of maximum entropy sequence prediction is an inference procedure which requires enumeration of all possible action sequences into the future given a set of observations. In the same way that the value function is computed for optimal control, the *log-partition function* of maximum entropy IOC can be computed using dynamic programming – differing only in the substitution of the "softmax" operator for the "max" operator in the Bellman equations. This relationship was noted as early as (Rust 1994) and formalized in (Ziebart et al. 2008). While dynamic programming renders this efficient for small scale problems, more appropriate techniques are needed for dealing with problems with large state spaces.

When the state space is large, one natural approach is to use approximate dynamic programming for the approximate calculation of these functions. We draw our inspiration from value function approximation methods, which have been successful in solving high-dimensional control problems (Tesauro 1994; Ernst, Geurts, and Wehenkel 2005; Riedmiller 2005; Mnih et al. 2013), to address the high-dimensional challenges in our scenario.

The algorithmic contribution of this work is an approximate MaxEnt IOC algorithm, suitable for dealing with high-dimensional problems, that uses an Approximate Value Iteration (AVI) algorithm to compute the softmax-based value (log-partition) function. The AVI procedure uses a regression estimator at each iteration, where the choice of the estimator is not constrained. In particular, we utilize a reproducing kernel Hilbert space-based (RKHS) regularized estimator due to its flexibility and favourable properties – though the framework is more general and allows other regression estimators such as local averagers, random forests, boosting, neural networks, etc. Efficient Monte Carlo sampling then enables a dimension-independent estimate of the gradient of the reward function.

The theoretical contribution of this paper is the analysis of this approximate procedure. We provide a finite-sample upper bound guarantee on the excess loss, i.e., the loss of our approximate procedure compared to an "ideal" MaxEnt IOC procedure without any approximation in the computation of the log-partition function or the feature expectation.

IOC for High-Dimensional Problems

The problem of the inverse optimal control (also known as inverse reinforcement learning) is to recover an agent's (or expert's) reward function given a controller or policy (or samples from the agent's behavior) when the dynamics of the process is known.

To describe our approach to IOC, which is based on the Maximum Entropy Inverse Optimal Control of (Ziebart, Bagnell, and Dey 2013), we first define a parametric-reward Markov Decision Process (θ-MDP). θ-MDP is defined as a tuple $(\mathcal{X}, \mathcal{A}, \mathcal{P}, \underline{g}, \theta)$, where $\mathcal{X}$ is a measurable state space (e.g., $\mathbb{R}^D$), $\mathcal{A}$ is a finite set of actions, $\mathcal{P} : \mathcal{X} \times \mathcal{A} \to \mathcal{M}(\mathcal{X})$ is the transition probability kernel, $\underline{g} : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^d$ is a mapping from state-action pairs to feature vectors of dimension d, and $\theta \in \mathbb{R}^d$ parametrizes the reward.[1] We consider θ-MDPs with finite horizon of T. For notational convenience, given a sequence $z_{1:T} = (z_1, \ldots, z_T)$, we denote $\underline{f}(z_{1:T}) = \sum_{t=1}^{T} \underline{g}(z_t)$. In IOC, we assume that $\mathcal{P}$ is known (or estimated separately).

Consider a set of demonstrated trajectories $\mathcal{D}_n = \{Z_{1:T}^{(i)}\}_{i=1}^n$ with each trajectory $Z_{1:T} = (Z_1, \ldots, Z_T) \sim \zeta$ with $Z_t = (X_t, A_t)$ and ζ being an unknown distribution over the set of trajectory. Also denote $\nu \in \mathcal{M}(\mathcal{X})$ as the distribution of X_1. We assume that this initial distribution is known. For a policy π, denote $P_\pi(Z_{1:T})$ as the distribution induced by following policy π. In the discrete state case, $P_\pi(Z_{1:T}) = \prod_{t=1}^{T-1} \mathcal{P}(X_{t+1}|X_t, A_t)\pi(A_t|X_t)$ (and similarly for continuous state spaces). Define the *causal conditioned probability* $\mathbb{P}\{A_{1:T}||X_{1:T}\} = \prod_{t=1}^{T} \mathbb{P}\{A_t|X_t\} = \prod_{t=1}^{T} \pi_t(A_t|X_t)$, which reflects the fact that future states do not influence earlier actions (compare with conditional probability $\mathbb{P}\{A_{1:T}|X_{1:T}\}$). We define the *causal entropy* H_π as $H_\pi = \mathbb{E}_{P_\pi(Z_{1:T})}[-\log \mathbb{P}\{A_{1:T}||X_{1:T}\}]$.

The primal optimization problem in Maximum Entropy Inverse Optimal Control estimator (Ziebart, Bagnell, and Dey 2013) is

$$\arg\max_{\pi} H_\pi\left(A_{1:T}||X_{1:T}\right) \tag{1}$$
$$\text{s.t.} \qquad \mathbb{E}_{P_\pi(Z_{1:T})}\left[\underline{f}(Z_{1:T})\right] = \frac{1}{n}\sum_{i=1}^{n} \underline{f}\left(Z_{1:T}^{(i)}\right).$$

The motivation behind this objective function is to find a policy π whose induced expected features, $\mathbb{E}_{P_\pi(Z_{1:T})}\left[\underline{f}(Z_{1:T})\right]$, matches the empirical feature count of the agent, that is $\frac{1}{n}\sum_{i=1}^{n} \underline{f}(Z_{1:T}^{(i)})$, while not committing to any distribution beyond what is implied by the data. The dual of this constrained optimization problem is (Theorem 3 of (Ziebart, Bagnell, and Dey 2013))

$$\min_{\theta \in \mathbb{R}^d} \log \mathcal{Z}_\theta - \left\langle \theta\,,\, \frac{1}{n}\sum_{i=1}^{n} \underline{f}\left(Z_{1:T}^{(i)}\right) \right\rangle, \tag{2}$$

in which $\log \mathcal{Z}_\theta$ is the log-partition function. For notational compactness, define $\hat{b}_n, \bar{b} \in \mathbb{R}^d$ as $\hat{b}_n = \frac{1}{n}\sum_{i=1}^{n} \underline{f}(Z_{1:T}^{(i)})$ and $\bar{b} = \mathbb{E}_{Z_{1:T}\sim\zeta}\left[\underline{f}(Z_{1:T})\right]$. The vector $\bar{b}$ is the true expected feature of the agent, which is unknown.

A key observation is that one might calculate $\log \mathcal{Z}_\theta$ using a Value Iteration (VI) procedure: For any $\theta \in \mathbb{R}^d$, define $r_t(x,a) = r(x,a) = \langle\, \theta\,,\, \underline{g}(x,a)\,\rangle$, and perform the following VI procedure: Set $Q_T = r_T$, and for $t = T-1, \ldots, 1$,

$$Q_t(x,a) = r_t(x,a) + \int \mathcal{P}(dy|x,a)V_{t+1}(y), \tag{3}$$
$$V_t(x) = \text{soft}\max(Q_t(x,\cdot) \triangleq \log\left(\sum_{a\in\mathcal{A}} \exp(Q_t(x,a))\right).$$

We compactly write $Q_t = r_t + \mathcal{P}^a V_{t+1}$, where $\mathcal{P}^a(\cdot|x) = \mathcal{P}(\cdot|x,a)$.

It can be shown that $\log \mathcal{Z}_\theta = \mathbb{E}_\nu\left[V_1(X)\right]$. Also the MaxEnt policy solution to (1), which is in the form of Boltzmann distribution, is $\pi_t(a|x) = \pi_{t,\theta}(a|x) = \frac{\exp(Q_t(x,a))}{\sum_{a'\in\mathcal{A}}\exp(Q_t(x,a))} = \exp(Q_t(x,a) - V_t(x))$.

Instead of (2), we aim to solve the following regularized dual objective

$$\min_{\theta\in\mathbb{R}^d} L(\theta, \hat{b}_n) \triangleq \log \mathcal{Z}_\theta - \left\langle\, \theta\,,\, \hat{b}_n\,\right\rangle + \frac{\lambda}{2}\left\|\theta\right\|_2^2, \tag{4}$$

which can be interpreted as a relaxation of the constraints in the primal as shown by (Dudík, Phillips, and Schapire 2004; Altun and Smola 2006). Adding a regularization has a Bayesian interpretation too, and corresponds to having a prior over parameters.

It can be shown that $\nabla_\theta \log \mathcal{Z}_\theta = \mathbb{E}_{P_\pi(Z_{1:T})}\left[\underline{f}(Z_{1:T})\right]$ with $X_1 \sim \nu$, so the gradient of the loss function, which can be used in a gradient-descent-like procedure, is

$$\nabla_\theta L(\theta, \hat{b}_n) = \mathbb{E}_{P_\pi(Z_{1:T})}\left[\underline{f}(Z_{1:T})\right] - \hat{b}_n + \lambda\theta \tag{5}$$

For problems with large state space, the exact calculation of the log-partition function $\log \mathcal{Z}_\theta$ is infeasible as is the calculation of the the expected features $\mathbb{E}_{P_\pi(Z_{1:T})}\left[\underline{f}(Z_{1:T})\right]$.

Nonetheless, one can aim to approximate the log-partition function and estimate the expected features. We use two key insights to design an algorithm that can handle large state spaces. The first is that one can approximate the VI procedure of (3) using function approximators. The Approximate Value Iteration (AVI) procedure has been successfully used and theoretically analyzed in the Approximate Dynamic Programming and RL literature (Ernst, Geurts, and Wehenkel 2005; Riedmiller 2005; Munos and Szepesvári 2008).

The second insight, which is also used in some previous work such as (Vernaza and Bagnell 2012), is that one can estimate an expectation by Monte Carlo sampling and the error behavior would be $O(\frac{1}{\sqrt{N}})$ (for N independent trajectories),

[1] $\mathcal{M}(\Omega)$ is the set of probability distributions over Ω.

Algorithm 1 – Backward pass

$\mathcal{D}_m^{(t)} = \{(X_i, A_i, R_i^t, X_i')\}_{i=1}^m$, $R_i^t = \langle\, \theta\,,\, \underline{g}(X_i, A_i)\,\rangle$
$\hat{Q}_T \leftarrow 0$
for $t = T-1, \ldots, 2, 1$ **do**
  $Y_i^t = R_i^t + \text{soft}\max \hat{Q}_{t+1}(X_i', \cdot)$
  $\hat{Q}_t \leftarrow \text{argmin}_Q \frac{1}{m}\sum_{i=1}^m \left|Q(X_i, A_i) - Y_i^t\right|^2 + \lambda_{Q,m}\|Q\|_{\mathcal{H}}^2$
  $\hat{\pi}_t(a|x) \propto \exp(\hat{Q}(x,a))$
end for

Algorithm 2 – Forward pass

$\underline{f} \leftarrow 0$
repeat
  $\hat{X}_1 \sim \nu$
  for $t = 1, \ldots T-1$ **do**
    $\hat{A}_t \sim \hat{\pi}_t(\cdot|\hat{X}_t)$, $\underline{f} \mathrel{+}= \underline{g}^t(\hat{X}_t, \hat{A}_t)$
    $\hat{X}_{t+1} \sim \mathcal{P}(\cdot|\hat{X}_t, \hat{A}_t)$
  end for
until N sample paths
$\underline{f} \leftarrow \frac{1}{N}\underline{f}$ (estimated log-partition function gradient)

which is a dimension-free rate. These procedures are summarized in Algorithms 1 and 2. We describe each of them in detail.

To perform AVI, we use samples in the form of $\mathcal{D}_m^{(t)} = \{(X_i, A_i, R_i, X_i')\}_{i=1}^m$ with $X_i \sim \eta \in \mathcal{M}(\mathcal{X})$, $A_i \sim \pi_b(X_i)$, $R_i \sim \mathcal{R}(\cdot|X_i)$, and $X_i' \sim \mathcal{P}(\cdot|X_i, A_i)$. Here π_b is a behavior policy.[2] Given these samples, one can estimate Q_t with $\hat{Q}_t$ by solving a regression problem in which the input variables are $Z_i = (X_i, A_i)$ and the target values are $R_i + \hat{V}_{t+1}(X_i')$, and $\hat{V}_{t+1} = \log\left(\sum_{a\in\mathcal{A}} \exp(\hat{Q}_t(x,a))\right)$. That is,

$$\hat{Q}_t \leftarrow \text{Regress}\left(\left\{\left((X_i, A_i), R_i + \hat{V}_{t+1}(X_i')\right)\right\}_{i=1}^m\right).$$

Let us define $\tilde{Q}_t = r_t + \mathcal{P}^a \hat{V}_{t+1}$ and note that $\mathbb{E}\left[R_i + \hat{V}_{t+1}(X_i')|(X_i, A_i)\right] = \tilde{Q}_t(X_i, A_i)$, i.e., $\tilde{Q}_t$ is the target regression function. We will shortly see that the quality of approximation, which is quantified by $\varepsilon_{\text{reg}}(t) \triangleq \|\hat{Q}_t - \tilde{Q}_t\|_2$, affects the excess error of approximate MaxEnt IOC procedure. One way to improve this error is by using powerful regression estimator such as the regularized least-squares estimators, similar to Regularized Fitted Q-Iteration (Farahmand et al. 2009):

$$\hat{Q}_t \leftarrow \operatorname*{argmin}_{Q\in\mathcal{F}^{|\mathcal{A}|}} \frac{1}{m}\sum_{i=1}^m \left|Q(X_i, A_i) - \left(R_i + \hat{V}_{t+1}(X_i')\right)\right|^2 + \lambda_{Q,m} J(Q).$$

Here $\mathcal{F}^{|\mathcal{A}|}$ is the set of action-value functions, $J(Q)$ is the regularization functional, which allows us to control the complexity, and $\lambda_{Q,m} > 0$ is the regularization coefficient. The regularizer $J(Q)$ measures the complexity of function Q. Different choices of $\mathcal{F}^{|\mathcal{A}|}$ and J lead to different notions of complexity, e.g., various definitions of smoothness, sparsity in a dictionary, etc. For example, $\mathcal{F}^{|\mathcal{A}|}$ could be a reproducing kernel Hilbert space (RKHS) and J its corresponding norm, i.e., $J(Q) = \|Q\|_{\mathcal{H}}^2$. The AVI procedure with the RKHS-based formulation is summarized in Algorithm 1. Note that one may use any other regression method in this algorithm, and the theory would still hold.

To estimate $\mathbb{E}_{P_\pi(Z_{1:T})}\left[\underline{f}(Z_{1:T})\right]$ we may use Monte Carlo sampling: Draw a sample state from the initial distribution ν and then follow the sequence of policies π_t and count the features along the trajectory. Repeat this procedure N times (Algorithm 2). Because of the approximation of AVI, we do not have Q_t and consequently π_t, so we use $\hat{Q}_t$ and its corresponding Boltzmann policy $\hat{\pi}_t$. Therefore, instead of finding $\hat{\theta}_n$ minimizing the loss, i.e., $\nabla_\theta L(\hat{\theta}_n, \hat{b}_n) = 0$, we find a $\tilde{\theta}_n$ that makes the following "distorted" gradient of loss zero:

$$\nabla_\theta \tilde{L}(\theta, \hat{b}_n) = \frac{1}{N}\sum_{i=1}^N \underline{f}\left(\hat{Z}_{1:T}^{(i)}\right) - \hat{b}_n + \lambda\theta, \qquad (6)$$

where $\hat{Z}_{1:T}^{(i)} \sim P_{\hat{\pi}}(Z_{1:T})$. This causes some error in the estimation of $\mathbb{E}_{P_\pi(Z_{1:T})}\left[\underline{f}(Z_{1:T})\right]$. Also note that we do not have the true expected feature $\bar{b}$, but only $\hat{b}_n$. We would like to compare the loss of our procedure, that is $L(\tilde{\theta}_n, \hat{b}_n)$, compared to the best possible loss assuming that the log-partition function could be solved exactly, the expectation was calculated exactly, and the true expected feature vector was available, i.e., $\min_{\theta\in\mathbb{R}^d} L(\theta, \bar{b})$. The appendix in the supplementary material is devoted to the analysis of these sources of error in the quality of the obtained solution. Here we only report the main result.

Before presenting the result, we require a few more definitions. For $\theta, b \in \mathbb{R}^d$, define $L(\theta, b) = \log \mathcal{Z}_\theta - \langle\, \theta\,,\, b\,\rangle + \frac{\lambda}{2}\|\theta\|_2^2$. Let $\theta^* \leftarrow \text{argmin}_{\theta\in\mathbb{R}^d} L(\theta, \bar{b})$ and $\tilde{\theta}_n$ be the solution of $\nabla_\theta \tilde{L}(\tilde{\theta}_n, \hat{b}_n) = 0$. We use $\|\underline{g}(z)\|_p$ $(1 \le p \le \infty)$ to denote the usual vector space l_p-norm and we define $\|\underline{g}\|_{p,\infty} = \sup_z \|\underline{g}(z)\|_p$. We also define the following concentrability coefficients, similar to (Kakade and Langford 2002; Munos 2007; Farahmand, Munos, and Szepesvári 2010).

Definition 1 (Concentrability Coefficient of the Future-State Distribution)**.** *Given $\mu_1, \mu_2 \in \mathcal{M}(\mathcal{X})$, $k \ge 0$, and an arbitrary sequence of policies $(\pi_i)_{i=1}^k$, let $\mu_1\mathcal{P}^{\pi_1}\ldots P^{\pi_k} \in \mathcal{M}(\mathcal{X})$ denote the future-state distribution obtained when the first state is distributed according to μ_1 and then we follow the sequence of policies $(\pi_i)_{i=1}^k$. Define $C_{\mu_1,\mu_2}(k) \triangleq \sup_{\pi_1,\ldots,\pi_k} \|\frac{\mathrm{d}(\mu_1\mathcal{P}^{\pi_1}\ldots P^{\pi_k})}{\mathrm{d}\mu_2}\|_\infty$.*

Theorem 1. *Fix $\delta > 0$. Suppose that the excess error of the regression estimate at each time step $t = 1, \ldots, T-1$ is upper bounded by $\varepsilon_{reg}(t) \ge \|\hat{Q}_t - \tilde{Q}_t\|_{2,2(\eta)}$. Choose an arbitrary $\mu \in \mathcal{M}(\mathcal{X})$. Define $\varepsilon^2 \triangleq \left\|\underline{g}\right\|_{1,\infty}^2 (T+1)\Big[\frac{|\mathcal{A}|^2}{4}\sum_{t=1}^{T-1}(T+1-t)^3 C_{\nu,\mu}^2(t-1)\sum_{k=0}^{T-t} C_{\mu,\eta}(k)\varepsilon_{reg}^2(t+k) + 4T\left(\frac{8\ln(2/\delta)}{N} + \frac{1}{N}\right)\Big]$. The excess loss is then upper*

[2] In general, the distribution η used for the regression estimator is different from ζ. Furthermore, for simplicity of presentation and analysis, we assume that η is fixed for all time steps, but this is not necessary. In practice one might choose to use $\mathcal{D}_m^{(t)} = \mathcal{D}_n^{(t)}$ extracted from the demonstrated trajectories $\mathcal{D}_n$.

bounded by

$$L(\tilde{\theta}_n, \bar{b}) - L(\theta^*, \bar{b}) \leq \frac{16 \left\|\underline{g}\right\|_{2,\infty}^2 T \left(\frac{16 \ln(2/\delta)}{n} + \frac{2}{n}\right)}{\lambda} + \frac{2\sqrt{2} \left\|\underline{g}\right\|_{2,\infty} \sqrt{T} \left(\sqrt{\frac{8 \ln(2/\delta)}{n}} + \frac{1}{\sqrt{n}}\right) \varepsilon}{\lambda} + \frac{\varepsilon^2}{2\lambda},$$

with probability at least $1 - \delta$.

Notice the effect of the number of demonstrated trajectories n and the value of ε on the excess loss $L(\tilde{\theta}_n, \bar{b}) - \min_\theta L(\theta, \bar{b})$. By increasing n, the first two terms in the upper bound decreases with a dominantly $O(\frac{\varepsilon}{\lambda\sqrt{n}})$ behavior. The value of ε depends on several factors including the regression errors $\varepsilon_{\text{reg}}(t)$, the number of Monte Carlo trajectories N used in the Forward pass, and the behavior of MDP characterized by the concentrability coefficients.

The regression error depends on the regression estimator we use, the number of samples m, and the intrinsic difficulty of the regression problem characterized by its smoothness, sparsity, etc. For instance, if the input space $\mathcal{X}$ is D-dimensional and the regression function is k-times smooth, i.e., it belongs to the Sobolev space $\mathbb{W}^k(\mathbb{R}^D)$, the error ε_{reg} of the optimal estimator has $O(m^{-\frac{k}{2k+D}})$ behavior. The regularized least-squares estimators can achieve optimal error rate for a large class of problems including Sobolev spaces and many RKHSs. More examples of these standard results in the statistical learning theory are reported by (Györfi et al. 2002; Steinwart and Christmann 2008). We would like to emphasize that the analysis here is not for a specific regression estimator and one may use decision trees, random forest, deep neural networks, etc. for the task of regression.

Mental Simulation of Human Interactions

We validate our approach in the context of analyzing dual-agent interactions from video, in which the actions of one person are used to predict the actions of another (Huang and Kitani 2014). The key idea is that dual-agent interactions can be modelled as an optimal control problem, where the actions of the initiating agent induces a cost topology over the space of reactive poses – a space in which the reactive agent plans an optimal pose trajectory. Therefore, IOC can be applied to recover this underlying reactive cost function, which allows us to simulate mental images of the reactive body pose.

A visualization of the setting is shown in Figure 1. As shown in the figure, the ground truth sequence contains both the true reaction sequence $q_{1:T} = (q_1, \ldots, q_T)$ on the left hand side (LHS) and the pose sequence of the initiating agent (obervation) $o_{1:T} = (o_1, \ldots, o_T)$ on the right hand side (RHS). At training time, n demonstrated interaction pairs $\{q_{1:T}^{(i)}\}_{i=1}^n$ and $\{o_{1:T}^{(i)}\}_{i=1}^n$ are provided to learn the reward model of human interaction. At test time, only the initiating actions on the RHS $o_{1:T}$ are observed, and we perform inference over the previously learned reactive model to obtain an optimal reaction sequence $x_{1:T}$.

We follow (Huang and Kitani 2014) and model dual-agent interaction as a MDP in the following way. We use a high-

Figure 1: Examples of ground truth, partial observation, and visual simulation over occluded regions.

dimensional HOG (Dalal and Triggs 2005) feature of an image patch around a person as our state (pose) representation (Figure 2). The HOG feature is weighted by the probability of the foreground to filter out the background. This results in a continuous vector of 819 dimensions (64×112 bounding box). The actions are defined as the transition between states (poses), which are *deterministic* because we assume humans have perfect control over their body and one action will deterministically bring the pose to the next state.

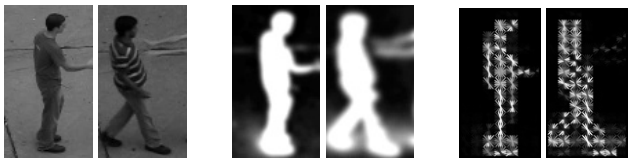
The features define the expressiveness of our cost function and are crucial to our method in modeling the dynamics of human interaction. We assume that the pose sequence $o_{1:T}$ of the initiating agent is observable on the RHS. For each frame t, we compute different features $\underline{g}_t(x, a) = (g_t^1, \ldots, g_t^d)$ from the sequence $o_{1:T}$. We modified the discrete features in (Huang and Kitani 2014) to adapt them to our approximate MaxEnt IOC for continuous state space.

Cooccurrence. Given a pose o_t on the RHS, we want to know how often a reactive pose x_t occurs on the LHS. This can be captured by the cooccurrence probability of poses on both LHS and RHS. We use kernel density estimation (Gaussian kernel with bandwidth 0.5) to approximate the cooccurrence probability $P_{co}(x, o)$ of LHS pose x and RHS pose o. Given a RHS pose o, we use the conditional probability $P_{co}(x|o)$ as our cooccurrence feature $g_t^1(x, a)$.

Transition. We want to know what actions will occur at a pose x, which model the probable transitions between consecutive states. Therefore, the second feature is the transition probability $g_t^2(x, a) = P_{tr}(x_a|x)$, where x_a is the state we will get to by performing action a at state x. Again, we use kernel density estimation to approximate $P_{tr}(x_a|x)$.

Smoothness. In addition to transition statistics from the training data, it is unlikely that the centroid position of human will change drastically between 2 frames and actions that induce high centroid velocity should be penalized. Therefore, we use the *smoothness* feature as $g_t^3(x, a) = 1 - \sigma(|v(x, a)|)$, where $\sigma(\cdot)$ is the sigmoid function, and $v(x, a)$ is the centroid velocity of action a at state x. These two features are independent of time step t.

Symmetry. In addition to the magnitude of centroid velocity, the relative velocity of the interacting agents is informative fir the current interaction. For example, in the hugging activity, the agents are approaching each other and will have a negative relative sign of centroid velocity. Therefore, we define two relative velocity features *attraction* and *repulsion* based on its sign. The feature *attraction* $g_t^4(x, a) = 1$ if and only if the interacting agents are moving in a symmetric way. We also define a complementary feature *repulsion* $g_t^5(x, a)$, which captures the case when the agents repel each other.



(a) Input image (b) Foreground map (c) HOG

Figure 2: 819-d HOG features (c) are weighted by the foreground map (b) and extracted from the original image (a).

Experiments

Given two people interacting, we observe only the actions of the initiator on the right hand side (RHS) and attempt to simulate the reaction on the left hand side (LHS). Since the ground truth distribution over all possible reaction sequences is not available, we measure how well the learned policy is able to describe the single ground truth pose sequence. For evaluation, we used videos from three datasets, *UT-interaction 1*, *UT-interaction 2* (Ryoo and Aggarwal 2010), and *SBU Kinect Interaction Dataset* (Yun et al. 2012) where the UTI datasets consist of only RGB videos, and SBU dataset consists of RGB-D (color plus depth) human interaction videos. In each interaction video, we occlude the ground truth reaction $q_{1:T} = (q_1, \dots, q_T)$ on the LHS, observe $o_{1:T} = (o_1, \dots, o_T)$ the action of the initiating agent on the RHS, and attempt to visually simulate $q_{1:T}$.

Metrics and Baselines

We compare the ground truth sequence with the learned policy using two metrics. The first one is probabilistic, which measures the probability of performing the ground truth reaction under the learned policy. A higher probability means the learned policy is more consistent with the ground truth reaction. We use the Negative Log-Likelihood (NLL):

$$-\log P(q_{1:T}|o_{1:T}) = -\sum_t \log P(q_t|q_{t-1}, o_{1:T}), \quad (7)$$

as our metric. In a MDP, $P(q_t|q_{t-1}, o_{1:T}) = \pi_{t-1}(a_{t-1}|q_{t-1})$, where the action a_{t-1} brings q_{t-1} to q_t. The second metric is deterministic, which directly measures the physical HOG distance (or joint distances for the skeleton video) of the ground truth reaction $q_{1:T}$ and the reaction simulated by the learned policy. The deterministic metric is the average image feature distance (AFD)

$$\frac{1}{T}\sum_t ||q_t - x_t||^2 \quad (8)$$

where x_t is the resulting reaction pose at frame t.

For model evaluation, we select four baselines to compare with the proposed method. The first baseline is the per frame nearest neighbor (NN) (Cover and Hart 1967), which only uses the *co-occurrence* feature at each frame *independently* and does not take into account the temporal context of states. For each observation o_t, we find the corresponding nearest LHS state with the highest cooccurrenceas $x_t^{NN} = \arg\max_x \hat{P}_{co}(x|o_t)$.

The second baseline is the hidden Markov model (HMM) (Rabiner and Juang 1986), which has been widely used to recover hidden time sequences given the observation. This fits our goal of simulating the hidden reactions given the observed actions of the initiating agent. HMM is defined by the transition probabilities $P(x_t|x_{t-1})$ and emission probabilities $P(o_t|x_t)$, which are equivalent to our *transition* and *cooccurrence* features. The weights for these two features are always the same in HMM, while our algorithm learns the optimal feature weights θ. We use the forward-backward algorithm to compute the likelihood. The optimal state sequence $x_{1:T}^{HMM}$ is computed by the Viterbi algorithm.

For the third baseline, we quantize the continuous state space into K discrete state by k-means clustering and apply the discrete Markov decision process (DMDP) inference used in (Kitani et al. 2012). The likelihood for MDP is computed by the stepwise product of the policy executions defined in (7).

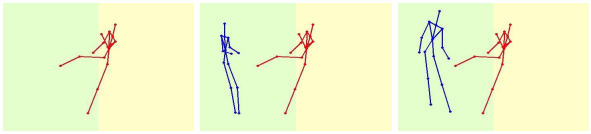
The forth baseline is the kernel-based reinforcement learning (KRL) (Ormoneit and Sen 2002) value function approximation presented in (Huang and Kitani 2014), which applies kernel regression on a value function learned by MaxEnt IOC to get a continuous value function over the whole state space. For a fair comparison for value function approximation we do not implement the extended mean-shift inference proposed in (Huang and Kitani 2014).

Table 1: AFD and NLL per activity category for UTI

AFD/NLL	NN	HMM	DMDP	KRL	Ours
shake	4.57/447	5.99/285	4.33/766	5.26/467	**4.08/213**
hug	4.78/507	8.89/339	**3.40**/690	4.11/475	3.53/**239**
kick	6.29/283	6.03/**184**	5.34/476	5.94/286	**3.92**/197
point	3.38/399	6.16/**321**	3.20/714	3.66/382	**3.06**/391
punch	3.81/246	5.85/193	4.06/396	4.71/254	**3.44/145**
push	4.21/315	7.73/214	**3.75**/446	4.67/324	3.94/**145**

Performance on 819-D HOG Space

We first evaluate our method on *UT-interaction 1*, and *UT-interaction 2* (Ryoo and Aggarwal 2010) datasets. The UTI datasets consists of RGB videos only, and some examples have been shown in Figure 1. The UTI datsets consist of 6 actions: hand shaking, hugging, kicking, pointing, punching, pushing. Each action has a total of 10 sequences for both datasets. We use 10-fold evaluation as in (Cao et al. 2013). We empirically set $K = 100$ for k-means and Gaussian kernel with bandwidth 0.5 for kernel density estimation. For the regression estimator in Backward pass (Algorithm 1), we use RKHS-based regularized least-squares estimator with a Gaussian kernel (equivalent to estimating the mean function of a Gaussian process with a Gaussian covariance kernel). We set $\lambda = \lambda_{Q,n} = 0.05$ as regularization coefficients. The average NLL and image feature distance per activity for each baseline is shown in Table 1. To evaluate the accuracy of our Monte Carlo (MC) sampling algorithm, we compare with the Forward pass in (Kitani et al. 2012) using our learned policy $\hat{\pi}$ ("Exact" in Table 1 and 2). Empirical results verify that our MC sampling strategy ($N = 500$) is able to achieve comparable performance. All optimal control based methods (DMDP and proposed) outperform the other two baselines in terms of image feature distance. Although the MDP is able to achieve a lower image feature distance than NN and HMM, the NLL is worse without proper regulariza-



Observation **Simulation** **Ground truth**

Figure 3: Observation, our simulation result, and the ground truth skeleton images of Kinect in SBU dataset.

tion. Furthermore, the proposed approximate MaxEnt IOC consistently outperforms the KRL value function approximation. Our method directly performs IOC on the continuous state space rather than interpolating value function of discretized state space.

Performance on 45-D Human Joint Space

We extend our framework to 3D human joint space. We evaluate our method on *SBU Kinect Interaction Dataset* (Yun et al. 2012), in which interactions performed by two people are captured by a RGB-D sensor and tracked skeleton positions at each frame are provided. In this case, the state space becomes a 15×3 (joint number times x, y, z) dimensional continuous vector. The SBU dataset consist of 8 actions: approaching, departing, kicking,pushing, shaking hands, hugging, exchanging object, punching. The first two actions (approaching & departing) are excluded from our experiments because the action of the initiating agent is to stand still and provides no information for forecasting. 7 participants performed activities in the dataset and results in 21 video sets, where each set contains videos of a pair of different people performing all interactions. We use 7-fold evaluation, in which videos of one participants are held out for one fold. The average NLL and AFD per activity are shown in Table 2. Again, the proposed model performs best. We note that in this lower-dimensional problem, the quantized model (DMDP) is able to achieve comparable performance.

Discussion

Our experiments demonstrate that it is possible to accurately mentally simulate (extrapolate images of body pose) using the IOC framework. The results are indicative of two important application domains that are enabled by this new framework: (1) anomaly detection and (2) reasoning about activities under heavy occlusion. Since the IOC framework can be used to simulate "typical" or expected sequential visualizations of human activity, they can be compared to observed activity to detect anomalous behavior. The same framework can be used to extrapolate a sequence of human poses even when a person might be fully occluded by exiting the field of the view of the camera or stand behind an obstruction.

The task of learning the underlying reward function of a Markov decision process from observed behavior has been studied as an inverse optimal control problem (Ziebart et al. 2008), also called inverse reinforcement learning (Abbeel and Ng 2004) or structural estimation (Rust 1994). In many approaches, parameters of the reward function are learned in an iterative procedure with repeated calls to a forward control or inference problem (Abbeel and Ng 2004; Ratliff, Bagnell, and Zinkevich 2006; Ziebart et al. 2008),

Table 2: AFD and NLL per activity category for SBU dataset

AFD/NLL	NN	HMM	DMDP	KRL	Ours (Exact)
kick	0.81/93	0.92/92	**0.65**/88	0.92/75	0.67/**58**
push	0.51/125	0.60/127	**0.45**/119	0.61/99	0.48/**78**
shake	0.48/149	1.41/151	0.42/145	0.54/121	**0.42/109**
hug	0.61/137	0.67/137	0.48/132	0.81/113	**0.47/96**
exchange	0.63/108	3.84/112	**0.53**/104	0.74/88	0.54/**72**
punch	0.56/98	0.66/99	**0.48**/93	0.66/78	0.52/**67**

though one may estimate the value function directly (Dvijotham and Todorov 2010) or solve a single large quadratic program (Ratliff, Bagnell, and Zinkevich 2006). The work of (Dvijotham and Todorov 2010), however, is developed for linearly-solvable MDPs, and more general MDPs should first be approximately embedded in the class of linearly-solvable MDPs. In addition, the rewards of linearly-solvable MDPs are assumed to be independent of actions.We follow Ziebart *et al*. (Ziebart et al. 2008; Ziebart, Bagnell, and Dey 2013), who formalized MaxEnt IOC, showing that the soft-maximum value function can be efficiently computed with dynamic programming for problems with finite state spaces.

Several approaches for inference and learning in high-dimensional problems have been proposed. Computational efficiency is straightforward for linear dynamical systems with quadratic costs (Ziebart 2010). (Dragan and Srinivasa 2012) leverage a related local quadratic approximation of the log-partition function for the forward problem. (Levine and Koltun 2012) learn local reward functions by considering a local linear-quadratic model. (Vernaza and Bagnell 2012) show that in the special case of continuous paths in $\mathbb{R}^D$ and the reward function of a high-dimensional problem possessing low-dimensional structure, a globally optimal solution can be attained. In contrast with these methods, our framework considers a **global** approximation and global reward learning not limited to continuous paths in $\mathbb{R}^D$ (admitting, e.g., discrete variables or stochastic dynamics) nor a low-dimensional reward constraint, and comes with finite-sample complexity guarantees.

Our formulation focuses on the prediction of decision, but similar model can also arise from information-theoretical constraints on decision making (Nilim and Ghaoui 2003; Todorov 2006; Theodorou and Todorov 2012). In this context, Monte Carlo sampling has been utilized in (Kappen 2005) to approximate the path integral computation, and function approximation of the desirability function has also been explored in (Todorov 2009; Zhong and Todorov 2011). The contribution of our work, however, lies in the combined application of these approaches to the context of learning a predictive model based on inverse reinforcement learning. Furthermore, we analyze this procedure and provide a finite-sample upper bound guarantee on the excess loss.

Acknowledgments

This research was sponsored in part by the Army Research Laboratory (W911NF-10-2-0061), the National Science Foundation (Purposeful Prediction: Co-robot Interaction via Understanding Intent and Goals), and the Natural Sciences and Engineering Research Council of Canada.

References

Abbeel, P., and Ng, A. Y. 2004. Apprenticeship learning via inverse reinforcement learning. In *ICML*.

Altun, Y., and Smola, A. 2006. Unifying divergence minimization and statistical inference via convex duality. In *COLT*.

Cao, Y.; Barrett, D. P.; Barbu, A.; Narayanaswamy, S.; Yu, H.; Michaux, A.; Lin, Y.; Dickinson, S. J.; Siskind, J. M.; and Wang, S. 2013. Recognize human activities from partially observed videos. In *CVPR*.

Cover, T., and Hart, P. 1967. Nearest neighbor pattern classification. *IEEE Trans. Information Theory* 13(1):21–27.

Dalal, N., and Triggs, B. 2005. Histograms of oriented gradients for human detection. In *CVPR*.

Dragan, A., and Srinivasa, S. 2012. Formalizing assistive teleoperation. In *RSS*.

Dudík, M.; Phillips, S. J.; and Schapire, R. E. 2004. Performance guarantees for regularized maximum entropy density estimation. In *COLT*, volume 3120. 472–486.

Dvijotham, K., and Todorov, E. 2010. Inverse optimal control with linearly-solvable mdps. In *ICML*.

Ernst, D.; Geurts, P.; and Wehenkel, L. 2005. Tree-based batch mode reinforcement learning. *JMLR* 6:503–556.

Farahmand, A.-m.; Ghavamzadeh, M.; Szepesvári, Cs.; and Mannor, S. 2009. Regularized fitted Q-iteration for planning in continuous-space Markovian Decision Problems. In *ACC*, 725–730.

Farahmand, A.-m.; Munos, R.; and Szepesvári, Cs. 2010. Error propagation for approximate policy and value iteration. In *NIPS*.

Györfi, L.; Kohler, M.; Krzyżak, A.; and Walk, H. 2002. *A Distribution-Free Theory of Nonparametric Regression*.

Huang, D.-A., and Kitani, K. M. 2014. Action-reaction: Forecasting the dynamics of human interaction. In *ECCV*.

Kakade, S., and Langford, J. 2002. Approximately optimal approximate reinforcement learning. In *ICML*, 267–274.

Kappen, H. J. 2005. Path integrals and symmetry breaking for optimal control theory. *Journal of statistical mechanics: theory and experiment* 2005(11).

Kitani, K. M.; Ziebart, B. D.; Bagnell, J. A.; and Hebert, M. 2012. Activity forecasting. In *ECCV*.

Levine, S., and Koltun, V. 2012. Continuous inverse optimal control with locally optimal examples. In *ICML*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing Atari with deep reinforcement learning. *CoRR* abs/1312.5602.

Munos, R., and Szepesvári, Cs. 2008. Finite-time bounds for fitted value iteration. *JMLR* 9:815–857.

Munos, R. 2007. Performance bounds in L_p norm for approximate value iteration. *SIAM Journal on Control and Optimization* 541–561.

Nilim, A., and Ghaoui, L. E. 2003. Robustness in Markov decision problems with uncertain transition matrices. In *NIPS*.

Ormoneit, D., and Sen, S. 2002. Kernel based reinforcement learning. *Machine Learning* 49(2-3):161–178.

Rabiner, L., and Juang, B.-H. 1986. An introduction to hidden Markov models. *ASSP Magazine*.

Ratliff, N. D.; Bagnell, J. A.; and Zinkevich, M. A. 2006. Maximum margin planning. In *ICML*.

Riedmiller, M. 2005. Neural fitted Q iteration – first experiences with a data efficient neural reinforcement learning method. In *ECML*, 317–328.

Rust, J. 1994. Structural estimation of Markov decision processes. *Handbook of econometrics* 4(4).

Ryoo, M. S., and Aggarwal, J. K. 2010. UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA). http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html.

Steinwart, I., and Christmann, A. 2008. *Support Vector Machines*. Springer.

Tesauro, G. 1994. TD-gammon, a self-teaching backgammon program, achieves master-level play. *Neural Computation* 6:215–219.

Theodorou, E., and Todorov, E. 2012. Relative entropy and free energy dualities: Connections to path integral and kl control. In *IEEE CDC*.

Todorov, E. 2006. Linearly-solvable Markov decision problems. In *NIPS*.

Todorov, E. 2009. Eigenfunction approximation methods for linearly-solvable optimal control problems. In *ADPRL*, 161–168. IEEE.

Vernaza, P., and Bagnell, J. A. 2012. Efficient high dimensional maximum entropy modeling via symmetric partition functions. In *NIPS*, 584–592.

Walker, J.; Gupta, A.; and Hebert, M. 2014. Patch to the future: Unsupervised visual prediction. In *CVPR*.

Yun, K.; Honorio, J.; Chattopadhyay, D.; Berg, T. L.; and Samaras, D. 2012. Two-person interaction detection using body-pose features and multiple instance learning. In *CVPRW*.

Zhong, M., and Todorov, E. 2011. Moving least-squares approximations for linearly-solvable stochastic optimal control problems. *Journal of Control Theory and Applications* 9(3):451–463.

Ziebart, B. D.; Bagnell, J. A.; and Dey, A. K. 2013. The principle of maximum causal entropy for estimating interacting processes. *IEEE Trans. on Information Theory* 59(4):1966–1980.

Ziebart, B. D.; Maas, A. L.; Bagnell, J. A.; and Dey, A. K. 2008. Maximum entropy inverse reinforcement learning. In *AAAI*, 1433–1438.

Ziebart, B. D. 2010. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Ph.D. Dissertation.