Inverse Optimal Heuristic Control for Imitation Learning

Nathan Ratliff, Brian Ziebart, Kevin Peterson, J. Andrew Bagnell, Martial Hebert, Anind K. Dey
Robotics Institute, MLD, CSD
Carnegie Mellon University
Pittsburgh, PA 15213

Siddhartha Srinivasa
Intel Research
Pittsburgh, PA 15213

Abstract

One common approach to imitation learning is *behavioral cloning* (BC), which employs straightforward supervised learning (i.e., classification) to directly map observations to controls. A second approach is *inverse optimal control* (IOC), which formalizes the problem of learning sequential decision-making behavior over long horizons as a problem of recovering a utility function that explains observed behavior. This paper presents *inverse optimal heuristic control* (IOHC), a novel approach to imitation learning that capitalizes on the strengths of both paradigms. It employs long-horizon IOC-style modeling in a low-dimensional space where inference remains tractable, while incorporating an additional descriptive set of BC-style features to guide a higher-dimensional overall action selection. We provide experimental results demonstrating the capabilities of our model on a simple illustrative problem as well as on two real world problems: turn-prediction for taxi drivers, and pedestrian prediction within an office environment.

1 Introduction

Imitation learning is an important tool for modeling decision-making agents (Ziebart *et al.*, 2008b) as well as for crafting automated decision making algorithms, particularly in robotics (Pomerleau, 1988; LeCun *et al.*, 2006; Ratliff *et al.*, 2006; Silver, Bagnell, & Stentz, 2008; Kolter, Abbeel, & Ng, 2007). The majority of imitation learning techniques rely on an approach we will term *behavioral cloning* (BC) (Bain & Sammut, 1995), in which one attempts to predict actions directly from an observed feature vector that describes the environment. This approach has been applied successfully in a number of applications. Perhaps the most well-known of these applications is the problem of learning to steer a vehicle within a road lane or around obstacles (Figure 1) using camera images (Pomerleau, 1988; LeCun *et al.*, 2006).

Figure 1: Left: Obstacles in the roadway that could influence a driver's actions in a behavioral cloning approach. Right: The inverse optimal control setting for navigating (cyan line) to a destination (red X). Driver behavior is best explained as a combination of these two approaches.

An alternate approach that has recently been successful for achieving coherent decision making– especially sequential decisions over multiple time-steps about distant future goals– has been to treat the imitation learning problem as one of *inverse optimal control* (IOC). Here the goal is to take training examples in the form of trajectories through the state space (Figure 1) and train a planning algorithm to predict the same set of decisions. Moreover, these algorithms train planning behavior that generalizes to new domains unseen during training. They demonstrate strong empirical and theoretical success on a wide variety of problems ranging from taxi route preference modeling and driver behavior modeling (Ziebart *et al.*, 2008a; Abbeel & Ng, 2004) to heuristic learning for legged locomotion and mobile robot navigation through rough terrain (Ratliff *et al.*, 2006; Ratliff, Bagnell, & Zinkevich, 2006; Silver, Bagnell, & Stentz, 2008; Kolter, Abbeel, & Ng, 2007).

In this work, we integrate these two previously disparate forms of imitation learning to engage the capabilities of both paradigms. The success of inverse optimal control centers around its ability to move beyond a one-step look-ahead paradigm and to learn a policy that correctly predicts entire sequences of actions. When applicable, these algorithms substantially outperform alternative techniques such as behavioral

Appearing in Proceedings of the 12^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2009, Clearwater Beach, Florida, USA. Volume 5 of JMLR: W&CP 5.

cloning. Unfortunately, as the dimensionality of the state space increases, the corresponding Markov Decision Process (MDP) becomes increasingly difficult to solve, severely limiting the application of these techniques. By contrast, behavioral cloning methods benefit from the scalability of existing supervised learning algorithms to handle high-dimensional feature spaces, but they ignore modern planning technologies and make decisions without explicitly considering their future consequences.

We frame the training of our combined model as an optimization problem. Although the resulting objective function is non-convex, we present a collection of convex approximations that may be optimized as surrogates. Further, we demonstrate both empirically and theoretically that the objective function is nearly convex. Optimizing it directly leads to improved performance across a range of imitation learning problems. Section 5 begins by illustrating the theoretical properties of our algorithm on a simple problem. We then demonstrate the algorithm and compare its performance to previous approaches on a taxi route prediction problem (Section 5.2) and on a pedestrian prediction problem (Section 5.3) using real-world data sets.

Prior work has previously examined combining aspects of behavioral cloning and inverse optimal control (under the names *direct* and *indirect* approaches) (Neu & Szepesvári, 2007), however, the authors focus on only the relationships between the loss-functions typically considered under the two approaches. The techniques described in that paper remain limited by the low-dimensionality restrictions of inverse optimal control. Formally, similar to previous work (Neu & Szepesvári, 2007; Ramachandran & Amir, 2007; Ziebart *et al.*, 2008a), our technique fits a Gibbs/Maximum Entropy model over actions based on features in the environment. In this paper, however, we take a direct approach to training and propose to use our Gibbs-based model to learn a stochastic policy that predicts the probability that the expert takes each action given an observation.

2 Inverse optimal heuristic control

In this section, we examine a relationship between behavioral cloning and inverse optimal control that becomes clear through the use of Gibbs distributions. After discussing this relationship in Section 2.1, we propose a novel Gibbs model in Section 2.2 that combines the strengths of these individual models. Section 2.3 presents an efficient gradient-based learning algorithm for fitting this model to training data.

2.1 Gibbs models for imitation learning

Recent research in inverse optimal control has introduced a Gibbs model of action selection in which the probability of taking an action is inversely proportional to that action's exponentiated Q-value in the MDP (Neu & Szepesvári, 2007; Ramachandran & Amir, 2007). Denoting the immediate cost of taking an action a from state s as $c(s,a)$, and the cost-to-go from a state s' as $J(s')$, we can write $Q^*(s,a) = c(s,a) + J(\mathcal{T}_s^a)$, where $\mathcal{T}_s^a$ denotes the deterministic transition function.[1] The Gibbs model is therefore

$$p(a|s) = \frac{e^{-c(s,a)-J(\mathcal{T}_s^a)}}{\sum_{a' \in \mathcal{A}_s} e^{-c(s,a')-J(\mathcal{T}_s^{a'})}}, \qquad (1)$$

where the function $Q^*(s,a) = c(s,a) + J(\mathcal{T}_s^a)$ is known as the energy function of the Gibbs model.

The form of this inverse optimal control model is strikingly similar to a multi-logistic regression classification model (Nigam, Lafferty, & McCallum, 1999). We arrive at a straight-forward behavioral cloning model based on multi-logistic regression by simply replacing the energy function with a linear combination of features: $E(s,a) = w^T f(s,a)$, where $f(s,a)$ denotes a function mapping each state-action pair to a representative vector of features.

2.2 Combining inverse optimal control and behavioral cloning

The observations presented in Section 2.1 suggest that a natural way to combine these two models is to design an energy function that utilizes the strengths of both paradigms. While inverse optimal control has demonstrated better generalization than behavioral cloning in real-world settings (Ratliff, Bagnell, & Zinkevich, 2006), the technique's applicability is limited by its reliance on an MDP solver. On the other hand, the range of application for behavioral cloning has historically surpassed the modeling capacity of MDPs.

To prevent needlessly restricting our model, we consider a general class of policies in which the agent simply maps observations to actions $p(a|o) \propto e^{-\widetilde{E}(o,a)}$, where $a \in \mathcal{A}$ is an arbitrary action and $o \in \mathcal{O}$ is a given observation.

In many cases, there exists a problem specific MDP $\mathcal{M}$ that can model a subproblem of the decision process. Let $\mathcal{S}_\mathcal{M}$ and $\mathcal{A}_\mathcal{M}$ be the state and action spaces of our lower-dimensional MDP $\mathcal{M}$. We require a mapping $\phi(o,a)$ from an observation-action pair to a sequence of state-action pairs that represents the behavior exhibited by the action through the lower-dimensional MDP. Specifically, we denote $\phi(o,a) = \{(s_1,a_1),(s_2,a_2),\dots\}$ and use $\mathcal{T}_o^a$ to indicate the state resulting from following the action-trajectory $\phi(o,a)$. Throughout the paper, we denote the cost of a trajectory ξ through this MDP as $C(\xi)$, although for convenience we often abbreviate $C(\phi(o,a))$ as $C(o,a)$.

Our combined Gibbs model, in this setting, uses the following energy function

$$\widetilde{E}(o,a) = E(o,a) + Q^*_\mathcal{M}(o,a),$$

[1] In this paper, we restrict ourselves to deterministic MDPs for modeling the lower-dimensional problem.

where $Q^*_{\mathcal{M}}(o,a) = C(o,a) + J_{\mathcal{M}}(\mathcal{T}^a_o)$ denotes the cumulative cost-to-go of taking the short action-trajectory $\phi(o,a)$ and then following the minimum cost path from the resulting state to the goal. Intuitively, the learning procedure chooses between BC and IOC paradigms under this model, or finds a combination of the two that better represents the data.

In what follows, we denote a trajectory as a sequence of observation-action pairs $\xi = \{(o_t, a_t)\}_{t=1}^{T_\xi}$, the set of all such trajectories starting from a given observation o as Ξ_o, and the set of actions that can be taken given a particular observation o as $\mathcal{A}_o$. Sections 3 and 4 present some theoretical results for the Gibbs IOC model. In those sections, we refer to only a single MDP, and we can talk about states in place of observations. When applicable, we replace the observation o with a state s in this notation.

Choosing linear parameterizations of both terms of the energy function we can write our model as

$$p(a|s) = \frac{e^{-w_v^T f_v(o,a) - w_c^T F^*_{\mathcal{M}}(o,a)}}{\sum_{a' \in \mathcal{A}} e^{-w_v^T f_v(o,a) - w_c^T F^*_{\mathcal{M}}(o,a)}}, \qquad (2)$$

where we define $F_{\mathcal{M}}(\xi) = \sum_t f_c(s_t, a_t)$ as the sum of the feature vectors encountered along trajectory ξ, and denote

$$F^*_{\mathcal{M}}(o,a) = F_{\mathcal{M}}(\phi(o,a)) + \sum_{(s_t,a_t) \in \xi^*} f_c(s_t, a_t),$$

where ξ^* is the optimal trajectory starting from $\mathcal{T}^a_o$ and $f_c(s,a)$ denotes a feature vector associated with state-action pair (s,a). Below we use w to denote the combined set of parameters (w_v, w_c). This notation utilizes the common observation that for linear costs, the cumulative cost-to-go of a trajectory through an MDP is a linear combination of the cumulative feature vector (Ng & Russell, 2000). Choosing $w_v = 0$ results in a generalized Gibbs model for IOC in which each action may be a small trajectory through the MDP.

We often call $f_v(s,a)$ and $f_c(s,a)$ *value features* and *cost features* respectively because of their traditional uses for value function approximation and cost parameterization in the separate behavioral cloning and inverse optimal control models.

2.3 Gradient-based optimization

Following the tradition of multi-logistic regression for behavioral cloning, given a trajectory $\xi = \{(o_t, a_t)\}_{t=1}^{T}$, we treat each observation-action pair as an independent training example and optimize the negative log-likelihood of the data. Given a set of trajectories $\mathcal{D} = \{\xi_i\}_{i=1}^N$, the exponential form of our Gibbs distribution allows us to write our objective $l(\mathcal{D}; w_v, w_c) = -\log \prod_{i=1}^N p(\xi_i)$ as

$$l(\mathcal{D}; w_v, w_c) = \sum_{i=1}^{N} \sum_{t=1}^{T_i - 1} w_v^T f_v(o_t, a_t) + w_c^T F^*_{\mathcal{M}}(o_t, a_t) + \log \sum_{a \in \mathcal{A}} e^{-w_v^T f_v(o_t,a) - w_c^T F^*_{\mathcal{M}}(o_t,a)} + \frac{\lambda}{2}\|w\|^2, \quad (3)$$

where we denote $T_i = T_{\xi_i}$ for convenience; $\lambda \geq 0$ is a regularization constant. In our gradient expressions and discussion below, we suppress the regularization term for notational convenience.

This objective function is piecewise-differentiable; at points of differentiability the following formula gives its gradient in terms of both w_v and w_c:

$$\nabla_{w_v} l(\mathcal{D}) = \sum_{i=1}^{N} \sum_{t=1}^{T_i - 1} f_v(o_t, a_t) - E_{p_w(a|o_t)}[f_v(o_t, a)]$$

$$\nabla_{w_c} l(\mathcal{D}) = \sum_{i=1}^{N} \sum_{t=1}^{T_i - 1} F^*_{\mathcal{M}}(o_t, a_t) - E_{p_w(a|o_t)}[F^*_{\mathcal{M}}(o_t, a)],$$

where we use $p_w(a|o) \propto \exp\{-w_v^T f_v(o,a) - w_c^T F^*_{\mathcal{M}}(s,a)\}$ to denote policy under our Gibbs model parameterized by the combined vector of parameters $w = (w_v; w_c)$. At points of nondifferentiability, there are multiple optimal paths through the MDP; choosing any one of them in the above formula results in a valid subgradient. Algorithm 1 presents a simple optimization routine based on exponentiated gradient descent for optimizing this objective.

3 On the efficient optimization of inverse optimal heuristic control

Our experiments indicate that the simple gradient-based procedure of Section 2.3 is robust to variations in starting point, a property often reserved for convex functions. This section demonstrates that, in many ways, our objective closely resembles a convex function. We first note that for any fixed w_c the objective as a function of w_v is convex. Moreover, we demonstrate below that for $w_v = 0$, the resulting (generalized) IOC Gibbs model is almost-convex in a rigorous sense. In combination, these results suggest that an effective strategy for optimizing our model is to:

1. Set $w_v = 0$ and optimize the almost-convex IOC Gibbs model.
2. Fix w_c and optimize the resulting convex problem in w_v.
3. Further optimize w_c and w_v jointly.

Optionally, in (2), one may use the fixed Q-values as features thereby guaranteeing that the resulting model can only improve over the Gibbs model found in step

Algorithm 1 Optimization of the negative log-likelihood via exponentiated gradient descent

procedure OPTIMIZE($\mathcal{D} = \{(\xi_i, \mathcal{M}_i)\}_{i=1}^N$)
  Initialize log-parameters $w_v^l \leftarrow 0$ and $w_c^l \leftarrow 0$
  for $k = 0, \ldots, K$ **do**
    Initialize $g_v = 0$ and $g_c = 0$
    for $i = 1, \ldots, N$ **do**
      Set $w_v = e^{w_v^l}$ and $w_c = e^{w_c^l}$
      Construct cost map $c_i^{sa} = w^T f_c(s, a)$ for our low-dimensional MDP $\mathcal{M}_i$
      Compute cumulative feature vectors $F^*_{\mathcal{M}}(o, a)$ through MDP $\mathcal{M}_i$ for each potential action from each observation found along the trajectory
      for $t = 1, \ldots, \mathcal{T}_i$ **do**
        $g_v^t \leftarrow f_v(o_t, a_t) - E_{p_w(a|o_t)}[f_v(o_t, a)]$
        $g_c^t \leftarrow F_{\mathcal{M}_i}(o_t, a_t) - E_{p_w(a|o_t)}[F^*_{\mathcal{M}_i}(o_t, a)]$
      end for
    end for
    Update $w_v^l \leftarrow w_v^l - \alpha_k \sum_{t=1}^{T_i} g_v^t$ and $w_c^l \leftarrow w_v^l - \alpha_k \sum_{t=1}^{T_i} g_c^t$
  end for
  return Final values $w_v = e^{w_v^l}$ and $w_c = e^{w_c^l}$
end procedure

(1). The final joint optimization phase can then only additionally improve over the model found in (2).

In what follows, we derive our almost-convexity results strictly in terms of the Gibbs model for IOC. Similar arguments can be used to prove analogous results for our generalized IOC Gibbs model as well.

Definition 3.1: Almost-convexity. *A function $f(x)$ is almost-convex if there exists a constant $c \in \mathbb{R}$ and a convex function $h(x)$ such that $h(x) \leq f(x) \leq h(x) + c$ for all x in the domain.*

The notion of almost-convexity formalizes the intuition that the objective function may exhibit the general shape of a convex function, while not necessarily being precisely convex. In what we show below, the negative log-likelihood of the Gibbs model for IOC is largely dominated by a commonly seen function in the machine learning literature known as the perceptron objective.[2] The nonconvexities of the negative log-likelihood arise from a collection of bounded discrepancy terms that measure the difference between the hard-min and the soft-min functions.

[2]We call this function the perceptron objective because the perceptron algorithm that has been used in the past for various structured prediction problems (Collins & Roark, 2004) is a particular subgradient algorithm for optimizing the function. We note, however, that technically, as an objective, this function is degenerate in the sense that it is successfully optimized by the zero function. Many of the properties of the perceptron algorithm cited in the literature are specific to that particular algorithm, and not general properties of this objective.

The negative log-likelihood of an example trajectory ξ, under this model is

$$-\log p(\xi) = -\sum_{t=1}^{T-1} \log \frac{e^{-Q^*(s_t,a_t)}}{\sum_{a\in\mathcal{A}_{s_t}} e^{-Q^*(s_t,a)}} = \sum_{t=1}^{T-1} \left(Q^*(s_t, a_t) + \log \sum_{a\in\mathcal{A}_{s_t}} e^{-Q^*(s_t,a)} \right). \quad (4)$$

By expanding $Q^*(s, a) = c(s, a) + J(\mathcal{T}_s^a)$, and then pushing the sum through, we can write this as

$$-\log p(\xi) = C(\xi) + \sum_{t=1}^{T-1} \left(J(\mathcal{T}_{s_t}^{a_t}) + \log \sum_{a\in\mathcal{A}_{s_t}} e^{-J(s_t,a)} \right),$$

where we denote the cumulative cost of a path as $C(\xi) = \sum_{t=1}^{T-1} c(s_t, a_t)$.

By noting that $\mathcal{T}_{s_t}^{a_t} = s_{t+1}$ (taking action a_t from state s_t gets us to the next state along the trajectory in a deterministic MDP), we rewrite the second term as

$$\sum_{t=1}^{T-1} J(\mathcal{T}_{s_t}^{a_t}) = \sum_{t=2}^{T} J(s_t) = -J(s_1) + \sum_{t=1}^{T-1} J(s_t).$$

For the final simplification, we added and subtracted $J(s_1) = \min_{\xi\in\Xi_{s_1}} C(\xi)$ and use the fact that the cumulative cost-to-go of the goal state s_T is zero (i.e., $J(s_T) = 0$). We additionally note that $J(s) = \min_{a\in\mathcal{A}_s} c(s, a) + J(\mathcal{T}_s^a)$. Our negative log-likelihood expression therefore simplifies to

$$-\log p(\xi) = C(\xi) - \min_{\xi\in\Xi_{s_1}} C(\xi) + \sum_{t=1}^{T-1} \left(\min_{a\in\mathcal{A}_{s_t}} Q^*(s_t, a) + \log \sum_{a\in\mathcal{A}_{s_t}} e^{-Q^*(s_t,a)} \right).$$

Finally, if we denote the soft-min function by $\min_{a\in\mathcal{A}_s}^s Q^*(s, a) = -\log \sum_{a\in\mathcal{A}_s} e^{-Q^*(s,a)}$, we can write the negative log-likelihood of a trajectory as

$$-\log p(\xi) = C(\xi) - \min_{\xi\in\Xi_{s_1}} C(\xi) + \sum_{t=1}^{T-1} \min_{a\in\mathcal{A}_{s_t}}^{\Delta} Q^*(s_t, a). \quad (5)$$

where we use the notation $\min_i^{\Delta} c_i = \min_i c_i - \min_i^s c_i$ to denote the discrepancy between the hard- and soft-min operators over a set of values $\{c_i\}$.

When the cost function is defined as a linear combination of features, the cost function itself is linear, and the Q^* function, as a min over linear functions, is therefore concave (making $-Q^*$ convex). Thus, the first two terms are both convex. In particular, they form the convex structured perceptron objective. Intuitively, these terms contrast the cumulative cost-to-go of the given trajectory with the minimum hypothesized cost-to-go over all trajectories.

The non-convexity of the negative log-likelihood objective arises from the final set of terms in Equation 5. Each of these terms simply denotes the difference between the soft-min function and the hard-min function. We can bound the absolute difference between the hard- and soft-min by $\log n$, where n is the number of elements over which the min operates. In our case, this means that each hard/soft-min discrepancy term can contribute no more to the objective than the constant value $\log |\mathcal{A}|$. In particular, we can state the following

Theorem 3.2: Gibbs IOC is almost-convex. *Denote the negative log-likelihood of ξ by $f(w)$ and let $h(w) = C(\xi) - \min_{\xi \in \Xi_{s_1}} C(\xi)$. If $n \geq |\mathcal{A}_s|$ for all s and $T = |\xi|$ is the length of the trajectory, then $h(w) \leq f(w) \leq h(w) + |\xi| \log n$.*

Proof. The soft-min is everywhere strictly less than the hard-min. Each discrepancy term $\min^{\Delta}_{a \in \mathcal{A}_{s_t}} Q^*(s_t, a)$ is therefore positive. Thus, $h(w) \leq f(w)$. Additionally, we know that $\min_i c_i - \log n \leq -\log \sum_i e^{-c_i}$ for any collection $\{c_i\}_{i=1}^n$. For a discrepancy term, this bound gives

$$\begin{aligned} \min{}^{\Delta}_i c_i &= \min_i c_i + \log \sum_i e^{-c_i} \\ &\leq \min_i c_i - (\min_i c_i - \log n) = \log n. \end{aligned}$$

Applying this upper bound to our objective gives the desired result. □

Section 5.1 presents some simple experiments illustrating these results.

4 Convex approximations

The previous section demonstrates that the generalized Gibbs model for inverse optimal control is almost-convex and suggests that directly optimizing the objective following algorithm 1 will work well on a wide range of problems. While our experimental results support this analysis, without a proof of (approximate) global convergence, we cannot guarantee these observations to hold across all problems. This section, therefore, presents three convex approximations to the negative log-likelihood that can be efficiently optimized to attain a good starting point for algorithm 2.3 should arbitrary initialization fail.

We present the first two results for the traditional IOC Gibbs model, although analogous results hold for our generalized Gibbs model as well. However, to emphasize its application to our full combined model, the discussion in Section 4.3 of what we call the soft-backup approximation is presented in full generality. In particular, steps (1) and (2) of the optimization strategy proposed in Section 3 may be replaced by optimizing this convex approximation.

4.1 The perceptron algorithm

Given the discussion of almost-convexity in Section 3, the simplest convex approximation is the perceptron objective

$$h(w) = \sum_{i=1}^{N} C(\xi_i) - \min_{\xi \in \Xi_i} C(\xi). \tag{6}$$

See (Collins & Roark, 2004) for details regarding the perceptron algorithm. Since each discrepancy term is positive, this approximation is a convex lower bound.

4.2 Expert augmentation

Our second convex approximation can be derived by augmenting how we compute the action probabilities of the Gibbs model. Equation 1 dictates that the probability of taking action a from state s_t is inversely proportional to $E(s_t, a) = Q^*(s_t, a)$ through our MDP. We will modify this energy function only for the value at the action a_t chosen by the expert from that state. Specifically, we will prescribe $E(s_t, a_t) = C(\xi_t) - \sum_{\tau=t}^{T} c(s_t, a_t)$; the energy of the expert's action now becomes the cumulative cost-to-go of the trajectory taken by the expert.

The negative log-likelihood of the resulting policy is convex. Moreover, for any parameter setting w, since $Q^*(s_t, a_t) \leq Q(\xi_t)$ and the energies of each alternative action remain unchanged, the probability of taking the expert's action cannot be smaller under the actual policy than it was under this modified policy. This observation shows that our modified negative log-likelihood forms a convex upper bound to the desired negative log-likelihood objective.

4.3 Soft-backup modification

Arguably, the most accurate convex approximation we have developed comes from deriving a simple soft-backup dynamic programming algorithm to modify the energy values for the expert's action used by the Gibbs policy. Applying this procedure backward along the example trajectory $\xi = \{(o_t, a_t)\}_{t=1}^{T}$ starting from the o_T and proceeding toward o_1 modifies the policy model in such a way that the contrasting hard/soft-min terms of Equation 5 cancel.

Specifically, as detailed in Algorithm 2 the soft-backup algorithm proceeds recursively from observation o_T replacing each hard-min J-value $J^*_{\mathcal{M}}(o_t) = \min_{a \in \mathcal{A}} C(o_t, a) + J_{\mathcal{M}}(\mathcal{T}^a_{o_t})$ with the associated soft-min. We define

$$\widetilde{J}_{\mathcal{M}}(o_t) = -\log \sum_{a \in \mathcal{A}} e^{-C(o_t, a) - \widetilde{J}_{\mathcal{M}}(\mathcal{T}^a_{o_t})}, \tag{7}$$

with $\widetilde{J}(\mathcal{T}^a_{o_t}) = J(\mathcal{T}^a_{o_t})$ when $a \neq a_t$. Our new policy along the example trajectory, therefore, becomes

$$\widetilde{p}(a|o_t) = \frac{e^{-E(o_t, a) - C(o_t, a) - \widetilde{J}_{\mathcal{M}}(\mathcal{T}^a_{o_t})}}{\sum_{a' \in \mathcal{A}_s} e^{-E(o_t, a') - C(o_t, a') - \widetilde{J}_{\mathcal{M}}(\mathcal{T}^{a'}_{o_t})}} \tag{8}$$

Algorithm 2 Soft-backup procedure for convex approximation

procedure SOFTBACKUP(ξ, $c : \mathcal{S} \rightarrow \mathcal{A}$)
  Compute cost-to-go values $J_{\mathcal{M}}(s)$ for s in $\mathcal{M}$ that can be reached by at most one action from any $o_t \in \xi$.
  Initialize $\widetilde{J}_{\mathcal{M}}(s_T) = 0$
  for $t = T-1, \ldots, 1$ **do**
    Set $\alpha_t = \sum_{a \in \mathcal{A} \setminus a_t} e^{-E(o_t,a)-C(o_t,a)-J_{\mathcal{M}}(\mathcal{T}^a_{s_t})}$
    Set $\widetilde{\alpha}_t = e^{-E(o_t,a_t)-C(o_t,a_t)-\widetilde{J}_{\mathcal{M}}(o_{t+1})}$
    Update $\widetilde{J}_{\mathcal{M}}(o_t) = -\log(\alpha_t + \widetilde{\alpha}_t)$
  end for
  return Updated J-values.
end procedure

Theorem 4.1 : *The negative log-likelihood of the modified policy of Equation 8 is convex and takes the form*

$$\widetilde{l}(w;\xi) = \sum_{(o_t,a_t)\in\xi} (E(o_t,a_t) + C(o_t,a_t)) - \widetilde{J}_{\mathcal{M}}(o_1).$$

Proof (sketch). The true objective function is given in Equation 3. Under the modified policy model, each energy value of the Gibbs model becomes $E(o,a) + C(o,a) + \widetilde{J}_{\mathcal{M}}(\mathcal{T}^a_o)$. In particular, for each t along the trajectory, the modified J-value in that expression cancels with the log-partition term corresponding to observation-action pair (o_{t+1}, a_{t+1}). Therefore, summing across all timesteps leaves only the energy and cost segment terms $\sum_t E(o_t,a_t)+C(o_t,a_t)$ and the first observation's modified J-value $\widetilde{J}(o_1)$. This argument demonstrates the form of the approximation. Additionally, since each backup operation is a soft-min performed over concave functions, each term $\widetilde{J}(o_t)$ is also concave. The term $-\widetilde{J}(o_1)$ is therefore convex, which proves the convexity of our approximation. □

Setting the value features to zero $w_v = 0$, we can additionally show that this final approximation $\widetilde{l}(w)$ is also bounded by the perceptron $h(w)$ in the same was as our negative log-likelihood objective $l(w)$. Moreover, $l(w) \leq \widetilde{l}(w)$ everywhere. Combining these bounds, we can state that for a given example trajectory ξ and for all w,

$$h(w) \;\leq\; l(w) \leq \widetilde{l}(w) \;\leq\; h(w) + |\xi| \log |A|. \qquad (9)$$

This observation suggests that the soft-backup convex approximation is potentially the tightest of the approximations presented here. Our experiments in Section 5.1 support this claim.

5 Experimental results

In this section, we first demonstrate our algorithm on a simple two-dimensional navigational problem to illustrate the overall convex behavior of optimizing the negative log-likelihood, and compare that to the performance of the convex approximations discussed in section 4. We then present two real-world experiments, and compare the performance of the combined model (with value features) to that of the Gibbs model alone (without value features).

5.1 An illustrative example

In this experiment, we implemented Algorithm 1 on the simple navigational problem depicted in the left-most panel of Figure 2 and compared its performance to that of each convex approximation presented in Section 4 using only the traditional IOC Gibbs model. We manually generated 10 training examples chosen specifically to demonstrate stochasticity in the behavior. Our feature set consisted of 14 randomly positioned two-dimensional radial basis features along with a constant feature. We set our regularization parameter to zero for this problem.

As we predicted in Section 4, the backup approximation performs the best on this problem. Although it converges to a suboptimal solution, the negative log-likelihood levels off and does not significantly increase from the minimum attained value in contrast to the behavior seen in the perceptron and replacement approximations. The center two panels show the cost functions learned for this problem using Algorithm 1 (center-left) and by optimizing the soft-backup approximation (center-right).

5.2 Turn Prediction for Taxi Drivers

We now apply our approach to modeling the route planning decisions of drivers so that we can predict, for instance, whether a driver will make a turn at the next stoplight. The imitation learning approach to this problem (Ziebart *et al.*, 2008a), which learns a cost function based on road network features, has been shown to outperform direct action modeling approaches (Ziebart *et al.*, 2008b), which estimate the action probabilities according to previous observation proportions (Simmons *et al.*, 2006; Krumm, 2008). However, computational efficiency in the imitation learning approach comes at a cost: the state space of the corresponding Markov Decision Process must be kept small. Table 1 shows how the number of states in the Markov decision process grows when paths of length K decisions are represented by the state. Previously, the model was restricted to using only the driver's last decision as the state of the Markov Decision Process to provide efficient inference.

As a consequence, given the driver's intended destination, decisions within the imitation learning model are assumed to be independent of the previous driving decisions that led the driver to his current intersection. We incorporate the following action value features that are functions of the driver's previous decisions to relax this independence assumption:

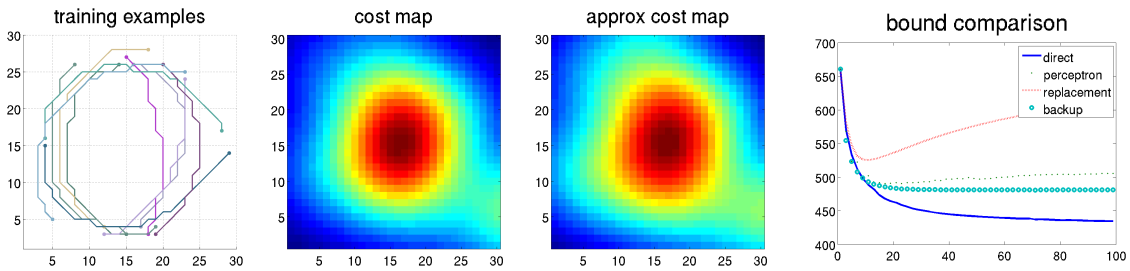

Figure 2: This figure shows the examples (left), the cost map learned by directly optimizing the negative log-likelihood (middle-left), the cost map learned by optimizing the soft-backup approximation (middle-right), and a plot comparing the performance of each convex approximation in terms of its ability to optimize the negative log-likelihood (right). The training examples were chosen specifically to exhibit stochasticity. See Section 5.1 for details.

Table 1: State space size for larger previous segment history.

History (K)	States
1	315704
2	901046
3	2496122
4	7733620
5	23281701
⋮	⋮
k	$\approx 3^k * 10^5$

- Has the driver previously driven on this road segment on this trip?
- Does this road segment lead to an intersection previously encountered on this trip?

We compare our approach with a combination of these action value features and additional cost-to-go features that are road network characteristics (e.g., length, speed limit, road category, number of lanes) with a model based on only cost-to-go features. Additionally, the *Full* model contains additional unique costs for every road segment in the network. We use the problem of predicting a taxi driver's next turn given final destination on a withheld dataset (over 55,000 turns, 20% of the complete dataset) to evaluate the benefits of our approach. We include baselines from previously applied approaches to this problem (Ziebart *et al.*, 2008b) and compare against the Gibbs model without value features, and our new inverse optimal heuristic control (IOHC) approach, which includes value features.

Table 2 shows the accuracy of each model's most likely prediction and the average log-probability of the driver's decisions within each model. We note for both sets of cost-to-go features a roughly 18% reduction in the turn prediction error over the Gibbs models when incorporating action value features (IOHC), which is statistically significant ($p < 0.01$). We also find corresponding improvements in our log-probability metric. We additionally note better improvement in both metrics over the best previously applied approaches for this task.

Table 2: Turn prediction evaluation for various models.

Model	Accuracy	Log Probability
Random Guess	46.4%	-0.781
Markov Model	86.2%	-0.319
MaxEnt IOC (Full)	91.0%	-0.240
Gibbs (Basic)	88.8%	-0.319
IOHC (Basic)	90.8%	-0.246
Gibbs (Full)	89.9%	-0.294
IOHC (Full)	**91.9%**	**-0.226**

Figure 3: The three images shown here depict the office setting in which the pedestrian tracking data was collected.

5.3 Pedestrian prediction

Predicting pedestrian motion is important for many applications, including robotics, home automation, and driver warning systems, in order to safely interact in potentially crowded real-world environments. Under the assumption that people move purposefully, attempting to achieve some goal, we can model a person's movements using an MDP and train a motion model using imitation learning.

In this experiment, we demonstrate that trajectories sampled from a distribution trained using momentum-based value features better match human trajectories than trajectories sampled from a model without value features. The resulting distribution over states is therefore a superior estimate of future behavior of the person being tracked.

Tracks of pedestrians were collected in an office environment using a laser-based tracker (see Figure 3). The outline of the room and the objects in the room

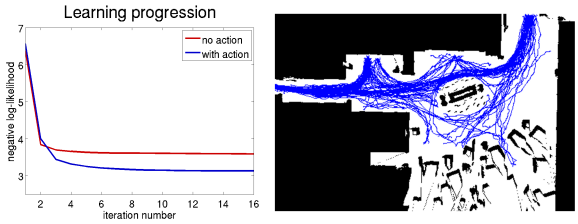

Figure 4: This figure compares the negative log-likelihood progressions between a traditional Gibbs model (without value features) and an IOHC model (with value features) on a validation set. Access to features encoding dynamic characteristics of the actions substantially improves modeling performance. The full collection of pedestrian trajectories is shown on the right from an overhead perspective.

were also recorded. The laser map was discretized into 15cm by 15cm cells and convolved with a collection of simple Gaussian smoothing filters. These filtered values and one feature representing the presence of an object make up the state-based feature set. Additionally, we include a set of action value features consisting of a history of angles between the current and previous displacements. These action features incorporate a smoothness objective that would otherwise require a higher-dimensional state space to incorporate.

We constructed a Gibbs model (without value features) and an IOHC model (with value features) using a set of 20 trajectories and we tested the model on a set of 20 distinct validation trajectories. Figure 4 compares the negative log-likelihood progression of both models on the test set during learning. The algorithm is able to exploit features that encode dynamical aspects of each action to find a superior model of pedestrian motion.

6 Conclusions

We have presented an imitation learning model that combines the efficiency and generality of behavioral cloning strategies with the long-horizon prediction performance of inverse optimal control. Our experiments have demonstrated empirically the benefits of this approach on real-world problems. In future work, we plan to explore applications of the pedestrian prediction model to developing effective robot-pedestrian interaction behaviors. In particular, since stochastic sampling may be implemented using efficient Monte Carlo techniques, the computational complexity of predictions can be controlled to satisfy real-time constraints.

Acknowledgments

The authors gratefully acknowledge the support of the DARPA Learning for Locomotion contract, the Quality of Life Technology Center/National Science Foundation contract under Grant No. EEEC-0540865, and Intel Research, Pittsburgh.

References

Abbeel, P., and Ng, A. Y. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proc. ICML*, 1–8.

Bain, M., and Sammut, C. 1995. A framework for behavioral cloning. In *Machine Intelligence Agents*, 103–129. Oxford University Press.

Collins, M., and Roark, B. 2004. Incremental parsing with the perceptron algorithm. In *Proc. ACL*, 111–118.

Kolter, J. Z.; Abbeel, P.; and Ng, A. 2007. Hierarchical apprenticeship learning with application to quadruped locomotion. In *Proc. NIPS*. 769–776.

Krumm, J. 2008. A markov model for driver route prediction. *Society of Automative Engineers (SAE) World Congress*.

LeCun, Y.; Muller, U.; Ben, J.; Cosatto, E.; and Flepp, B. 2006. Off-road obstacle avoidance through end-to-end learning. In *Proc. NIPS*. 739–746.

Neu, G., and Szepesvári, C. 2007. Apprenticeship learning using inverse reinforcement learning and gradient methods. In *Proc. UAI*, 295–302.

Ng, A. Y., and Russell, S. 2000. Algorithms for inverse reinforcement learning. In *Proc. ICML*, 663–670.

Nigam, K.; Lafferty, J.; and McCallum, A. 1999. Using maximum entropy for text classification. In *IJCAI-99 Workshop on Machine Learning for Information Filtering*.

Pomerleau, D. A. 1988. Alvinn: an autonomous land vehicle in a neural network. In *Proc. NIPS*, 305–313.

Ramachandran, D., and Amir, E. 2007. Bayesian inverse reinforcement learning. In *Proc. IJCAI*, 2586–2591.

Ratliff, N.; Bagnell, J. A.; and Zinkevich, M. 2006. Maximum margin planning. In *Proc. ICML*, 729–736.

Ratliff, N.; Bradley, D.; Bagnell, J. A.; and Chestnutt, J. 2006. Boosting structured prediction for imitation learning. In *Proc. NIPS*, 1153–1160.

Silver, D.; Bagnell, J. A.; and Stentz, A. 2008. High performance outdoor navigation from overhead data using imitation learning. In *Proceedings of Robotics Science and Systems*.

Simmons, R.; Browning, B.; Zhang, Y.; and Sadekar, V. 2006. Learning to predict driver route and destination intent. *Proc. Intelligent Transportation Systems Conference* 127–132.

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

Ziebart, B.; Maas, A.; Dey, A.; and Bagnell, J. A. 2008b. Navigate like a cabbie: Probabilistic reasoning from observed context-aware behavior. In *Proc. UbiComp*, 322–331.