Hierarchical Bayesian Inverse Reinforcement Learning

Jaedeug Choi, *Student Member, IEEE*, and Kee-Eung Kim, *Member, IEEE*

***Abstract*—Inverse reinforcement learning (IRL) is the problem of inferring the underlying reward function from the expert's behavior data. The difficulty in IRL mainly arises in choosing the best reward function since there are typically an infinite number of reward functions that yield the given behavior data as optimal. Another difficulty comes from the noisy behavior data due to sub-optimal experts. We propose a hierarchical Bayesian framework, which subsumes most of the previous IRL algorithms as well as models the sub-optimality of the expert's behavior. Using a number of experiments on a synthetic problem, we demonstrate the effectiveness of our approach including the robustness of our hierarchical Bayesian framework to the sub-optimal expert behavior data. Using a real dataset from taxi GPS traces, we additionally show that our approach predicts the driving behavior with a high accuracy.**

***Index Terms*—Decision theory, inverse problems, maximum a posteriori estimation.**

I. Introduction

INVERSE reinforcement learning (IRL) aims to determine the expert's underlying reward function from her behavior data and the dynamics model of environment [1]. IRL addresses the fundamental problem of finding the reward function in building a computational model for sequential decision making. It is a promising framework for examining animal and human behaviors [2] since the reward function represents an objective or a preference of the decision maker. As such, IRL has been applied to problems from various domains. Abbeel *et al.* [3] built controllers for helicopters to perform difficult aerobatic maneuvers utilizing human experts' demonstrations. Ziebart *et al.* [4] inferred taxi drivers' preferences from their GPS navigation data to predict their routes. Erkin *et al.* [5] estimated patients' preferences to determine the optimal timing of living-donor liver transplants. Chandramohan *et al.* [6] developed simulated users to assess the quality of dialogue management systems. Lee and Popović [7] constructed motion controllers to generate computer animation from exemplar motions. Vogel *et al.* [8] optimized the fuel efficiency of hybrid cars by implementing an driving route prediction system based on IRL.

In IRL, it is generally assumed that the expert acts in an environment modeled as a Markov decision process (MDP). Under the MDP formalism, the IRL problem is defined as finding the reward function that the expert is optimizing given the behavior data of state-action histories and the environment model of state transition probabilities. One of the inherent challenges of IRL is its ill-posedness, due to the fact that there are an infinite number of reward functions that arise as a valid solution. Hence, most of the studies on IRL have been devoted to defining an appropriate objective function that is used to single out the most meaningful reward function. Ng and Russell [9] searched for the reward function that maximizes the difference in the values of the expert's policy and the second best policy. Ratliff *et al.* [10] applied the structured max-margin optimization to find the reward function that maximizes the margin between the expert's policy and all other policies. Neu and Szepesvári [11] provided an algorithm for finding the policy that minimizes the deviation from the behavior. Ziebart *et al.* [12] adopted the principle of the maximum entropy for learning the policy whose feature expectations are constrained to match those of the expert's behavior. Ramachandran and Amir [13] presented a Bayesian approach formulating the reward preference as the prior and the behavior compatibility as the likelihood, and find the posterior mean of the reward function.

Although a remarkable progress has been made for the last decade exemplified by the above studies, they generally assume that the expert is a rational decision maker who can deduce all possible outcomes of actions for choosing the best decision. Under the MDP formalism, this assumption corresponds to the expert who always executes optimal actions derived from the optimal value function computed from the underlying reward function [9], [10], [14]. However, it is well known that human beings (and other animals) do not always exhibit completely rational behaviors, yielding sub-optimal action because of limitations in time, knowledge, and computational capabilities [15]–[17].

The motivation of our work is to address the behavior data with sub-optimal actions. Although there are a number of IRL algorithms capable of handling the sub-optimal behavior data, they treat it implicitly or use a fixed parameter value. Syed and Schapire [18] proposed a game-theoretic apprenticeship learning algorithm that aims to improve on the potentially sub-optimal expert. Ramachandran and Amir [13]

Manuscript received June 8, 2013; revised November 27, 2013, April 3, 2014, and June 30, 2014; accepted July 2, 2014. Date of publication October 1, 2014; date of current version March 13, 2015. This work was supported in part by the National Research Foundation of Korea under Grant NRF-2012R1A1A2007881 and in part by the IT Research and Development Program of MKE/KEIT under Contract 10041678. This paper was recommended by Associate Editor S. E. Shimony. *(corresponding author: K.-E. Kim.)*

J. Choi, deceased, was with the Department of Computer Science, Korea Advanced Institute of Science and Technology, Daejeon 305-701, Korea (e-mail: jdchoi@ai.kaist.ac.kr).

K.-E. Kim is with the Department of Computer Science, Korea Advanced Institute of Science and Technology, Daejeon 305-701, Korea (e-mail: kekim@cs.kaist.ac.kr).

Digital Object Identifier 10.1109/TCYB.2014.2336867

2168-2267 © 2014 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

and Neu and Szepesvári [11] used the softmax distribution over actions with a fixed value for the temperature parameter to represent the degree of sub-optimality. Ziebart *et al.* [12] used the softmax distribution over trajectories with the same parameterization. Rothkopf and Dimitrakakis [19] and Ramachandran and Amir [13] took a similar approach. As shown in the later section, the learning accuracy is sensitive to the selection of parameter values. Hence, it is desirable for IRL to be robust against a poor selection of parameter values.

The main contribution of this paper is to present a hierarchical Bayesian approach that:

1) explicitly learns the degree of sub-optimality in the behavior data by imposing a prior on the corresponding parameter;
2) makes the solution less affected by the prior on the reward function by imposing a hyper-prior.

Our approach is a hierarchical extension of our own previous work on Bayesian IRL [20] in the above two aspects, which allows the derivation of analytical formulas for computing gradients that are used for finding the MAP estimate of the reward function. We also report experimental results of applying our technique to the route prediction problem using a real dataset.

The rest of this paper is organized as follows. We start with a brief background on IRL (Section II). Next, we provide a detailed review on the Bayesian approach to IRL (Section III) since our work presented in this paper heavily builds on it. We then present the hierarchical Bayesian framework for IRL (Section IV), which is the main part of the paper. We report experimental results on a number of synthetic problems that are used as standard benchmark problems in IRL, and on the route prediction problem using a real dataset of GPS traces collected from taxi drivers [21] (Section V). Finally, we conclude the paper with discussions on our approach and directions for future work (Section V). All detailed proofs are provided in the appendix.

II. Preliminaries

A. Markov Decision Processes (MDPs)

A Markov decision process (MDP) [22] provides a mathematical framework for modeling a sequential decision making problem under uncertainty about the effect of an agent's action in an completely observable environment.

An MDP is defined as a tuple $\langle \mathcal{S}, \mathcal{A}, P_0, T, R, \gamma \rangle$ where

- $\mathcal{S}$ the finite set of states $s = 1, \ldots, |\mathcal{S}|$;
- $\mathcal{A}$ the finite set of actions $a = 1, \ldots, |\mathcal{A}|$;
- P_0 the starting state probability where $P_0(s)$ denotes the probability of starting in state s;
- T the state transition function where $T(s, a, s') = P(s'|s, a)$ denotes the probability of reaching state s' by taking action a in state s;
- R the reward function where $R(s, a)$ denotes the immediate reward of taking action a in state s, whose absolute value is bounded by $R_{\max}$;
- γ the discount factor whose value is in $[0, 1)$.

We use matrix notations to denote the transition function by an $|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}|$ matrix $\boldsymbol{T}$, and the reward function by an $|\mathcal{S}||\mathcal{A}|$-dimensional vector $\boldsymbol{r}$.

A policy in MDP is defined as a mapping $\pi : \mathcal{S} \to \mathcal{A}$, where $\pi(s) = a$ denotes the action to execute in state s is a.[1] The value of policy π is the expected discounted cumulative rewards obtained during following the policy and defined as:

$$V^{\pi} = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)|\pi\right]$$

where action a_t is chosen by policy π in state s_t. The state-value function of policy π for state s is computed by

$$V^{\pi}(s) = R(s, \pi(s)) + \gamma \sum_{s' \in \mathcal{S}} T\left(s, \pi(s), s'\right) V^{\pi}\left(s'\right) \quad (1)$$

so that $V^{\pi} = \sum_s P_0(s) V^{\pi}(s)$. Similarly, the action-value function (often called Q-function) is defined as

$$Q^{\pi}(s, a) = R(s, a) + \gamma \sum_{s' \in \mathcal{S}} T\left(s, a, s'\right) V^{\pi}\left(s'\right). \quad (2)$$

We can rewrite the above equations using matrix notations

$$\boldsymbol{v}^{\pi} = \boldsymbol{r}^{\pi} + \gamma \boldsymbol{T}^{\pi} \boldsymbol{v}^{\pi}$$
$$\boldsymbol{q}^{\pi}_{:,a} = \boldsymbol{r}^{a} + \gamma \boldsymbol{T}^{a} \boldsymbol{v}^{\pi}$$

where

- $\boldsymbol{T}^{\pi}$ an $|\mathcal{S}| \times |\mathcal{S}|$ matrix with $T^{\pi}_{s,s'} = T(s, \pi(s), s')$;
- $\boldsymbol{T}^{a}$ an $|\mathcal{S}| \times |\mathcal{S}|$ matrix with $T^{a}_{s,s'} = T(s, a, s')$;
- $\boldsymbol{r}^{\pi}$ an $|\mathcal{S}|$-dimensional vector with $r^{\pi}_{s} = R(s, \pi(s))$;
- $\boldsymbol{r}^{a}$ an $|\mathcal{S}|$-dimensional vector with $r^{a}_{s} = R(s, a)$;
- $\boldsymbol{v}^{\pi}$ an $|\mathcal{S}|$-dimensional vector with $v^{\pi}_{s} = V^{\pi}(s)$;
- $\boldsymbol{q}^{\pi}_{:,a}$ an $|\mathcal{S}|$-dimensional vector with $q^{\pi}_{s,a} = Q^{\pi}(s, a)$.

Additionally, we denote an $|\mathcal{S}||\mathcal{A}|$-dimensional vector $\boldsymbol{q}^{\pi} = [(\boldsymbol{q}^{\pi}_{:,1})^{\intercal}, \ldots, (\boldsymbol{q}^{\pi}_{:,|\mathcal{A}|})^{\intercal}]^{\intercal}$.

The agent's objective is to find an optimal policy π^* that maximizes the value for all the states. In other words, π^* is an optimal policy if and only if for all $s \in \mathcal{S}$

$$\pi(s) = \operatorname*{argmax}_{a \in \mathcal{A}} Q^{\pi^*}(s, a).$$

We denote $V^* = V^{\pi^*}$ and $Q^* = Q^{\pi^*}$.

The behavior data of policy π is defined to be the set $\Xi = \{\xi_1, \ldots, \xi_M\}$ of M trajectories by executing the policy, where the m-th trajectory ξ_m is an H-step sequence of state-action pairs: $\xi_m = \{(s_{m,1}, a_{m,1}), \ldots, (s_{m,H}, a_{m,H})\}$.[2] Given the set of trajectories, the value of the policy π can be empirically estimated by

$$\hat{V}^{\pi} = \frac{1}{M} \sum_{m=1}^{M} \sum_{h=1}^{H} \gamma^{h-1} R(s_{m,h}, a_{m,h}).$$

This formula will be used for estimating the value of the expert's policy π_E in IRL, since its behavior data is given instead of π_E.

For the rest of the paper, when we refer to some function f that is computed using the reward function $\boldsymbol{r}$, we use the notation $f(\boldsymbol{r})$ or $f(x; \boldsymbol{r})$ in order to be explicit. For example, we use the notation $V^{\pi}(s; \boldsymbol{r})$ to denote the value of policy π for state s using the reward function $\boldsymbol{r}$.

[1] This kind of policies are called deterministic policies. In some places, we slightly abuse the notation and use $\pi(s, a)$ for the probability of executing action a in state s for a stochastic policy π.

[2] Although we assume that all trajectories are of length H for notational brevity, our formulation extends to different lengths without modification.

B. IRL and Reward Optimality Condition

The IRL problem in MDP is formally stated as follows: given an MDP\R $\langle \mathcal{S}, \mathcal{A}, P_0, T, \gamma \rangle$ and the behavior data Ξ of the expert's policy π_E, find the reward function $\boldsymbol{r}$ that makes π_E an optimal policy for the given MDP.

Ng and Russell [9] presented a necessary and sufficient condition for reward function $\boldsymbol{r}$ of an MDP to guarantee the optimality of policy π

$$\boldsymbol{q}^{\pi}_{:,a}(\boldsymbol{r}) \leq \boldsymbol{v}^{\pi}(\boldsymbol{r}), \text{ for } \forall a \in \mathcal{A}.$$

From the condition, we obtain the following corollary, which is a succinct reformulation of the theorem by Ng and Russell [9].

Corollary 1: Given an MDP\R $\langle \mathcal{S}, \mathcal{A}, T, \gamma \rangle$, policy π is optimal if and only if reward function $\boldsymbol{r}$ satisfies

$$\left[\boldsymbol{I} - \left(\boldsymbol{I}^{\mathcal{A}} - \gamma \boldsymbol{T}\right)\left(\boldsymbol{I} - \gamma \boldsymbol{T}^{\pi}\right)^{-1} \boldsymbol{E}^{\pi}\right] \boldsymbol{r} \leq \boldsymbol{0} \tag{3}$$

where $\boldsymbol{E}^{\pi}$ is an $|\mathcal{S}| \times |\mathcal{S}||\mathcal{A}|$ matrix with the $(s, (s', a'))$ element being 1 if $s = s' \wedge \pi(s') = a'$ and 0 otherwise, and $\boldsymbol{I}^{\mathcal{A}}$ is an $|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}|$ matrix constructed by stacking the $|\mathcal{S}| \times |\mathcal{S}|$ identity matrix $|\mathcal{A}|$ times.

We refer to (3) as the reward optimality condition with respect to policy π. We also refer to the region bounded by (3) as the reward optimality region with respect to policy π since the set of linear inequalities in (3) defines the region of the reward functions that makes policy π optimal. We note that there are infinitely many reward functions in the reward optimality region even including constant reward functions (e.g., $\boldsymbol{r} = c\boldsymbol{1}$ where $c \in [-R_{\max}, R_{\max}]$). In other words, we have an infinite number of reward functions to choose from, including the degenerate ones. In order to resolve this nonuniqueness in solutions, IRL algorithms in the literature use a number of different preferences on reward functions.

C. Linearly Parameterized Reward Functions

When the state space is large, the reward function is often linearly parameterized as

$$R(s, a; \boldsymbol{r}) = r_1 \phi_1(s, a) + \ldots + r_D \phi_D(s, a) = \boldsymbol{r}^{\intercal} \boldsymbol{\phi}(s, a) \tag{4}$$

with predefined domain-dependent feature functions $\boldsymbol{\phi}$: $\mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}^D$ and the reward weight vector $\boldsymbol{r} = [r_1, \ldots, r_D]^{\intercal} \in \mathbb{R}^D$. Note that if we use $|\mathcal{S}||\mathcal{A}|$ indicator functions as features, one for each state-action pair, the reward function is represented as an $|\mathcal{S}||\mathcal{A}|$-dimensional vector as in the classical definition. We thus regard the reward function R equivalent to the D-dimensional reward vector $\boldsymbol{r}$ for the remainder of this paper since the classical definition is a special case of (4). Linear parameterization of the reward function is a common practice in IRL for large-scale MDPs [9], [14], [18].

III. Review on Bayesian IRL

Ramachandran and Amir [13] proposed a Bayesian framework for IRL (BIRL), where the prior encodes the preference on the reward function and the likelihood presents the compatibility of the reward function with the behavior data.

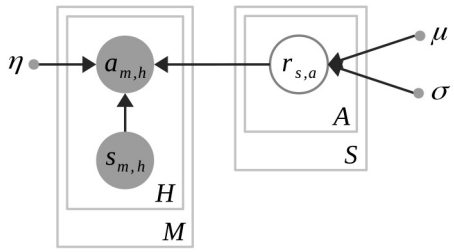


Fig. 1. Graphical representation of the BIRL model.

Assuming the rewards are i.i.d., the prior in BIRL is defined as

$$P(\boldsymbol{r}) = \prod_{i=1}^{D} P(r_i). \tag{5}$$

We can use various distributions as the prior. For example, the uniform prior can be used if we have no knowledge about the reward function other than its range, and the Gaussian or Laplacian distributions can be used if we prefer rewards to be close to some specific values. The Beta distribution can also be used if we treat rewards as the parameter of the Bernoulli distribution, i.e., $P(\chi_i = 1) = r_i$ using an auxiliary binary random variable χ_i [23].

The likelihood in BIRL is defined as an independent exponential, or softmax, distribution over actions

$$\begin{aligned} P(\Xi|\boldsymbol{r}, \eta) &= \prod_{m=1}^{M} P(\xi_m|\boldsymbol{r}, \eta) \\ &= \prod_{m=1}^{M} \prod_{h=1}^{H} \psi\left(a_{m,h}|s_{m,h}; \boldsymbol{r}, \eta\right) \end{aligned} \tag{6}$$

where $\psi(a|s; \boldsymbol{r}, \eta) = \frac{\exp(\eta q^*_{s,a}(\boldsymbol{r}))}{\sum_{a' \in \mathcal{A}} \exp(\eta q^*_{s,a'}(\boldsymbol{r}))}$ with η being the parameter representing the confidence of choosing optimal actions, and $\boldsymbol{q}^*(\boldsymbol{r})$ denotes the optimal Q-function computed from the MDP with reward function $\boldsymbol{r}$.

Using the Bayes theorem, we formulate the posterior over the reward function by combining the prior and the likelihood

$$P(\boldsymbol{r}|\Xi, \eta) \propto P(\Xi|\boldsymbol{r}, \eta) P(\boldsymbol{r}). \tag{7}$$

Fig. 1 shows the graphical model representing BIRL when the rewards are normally distributed with mean μ and standard deviation σ, that is

$$P(\boldsymbol{r}|\mu, \sigma) = \prod_{i=1}^{D} \mathcal{N}(r_i; \mu, \sigma). \tag{8}$$

Ramachandran and Amir [13] proposed a Markov chain Monte Carlo (MCMC) algorithm to compute the posterior mean of the reward function in BIRL, defined as

$$\boldsymbol{r}_{\text{MEAN}} = \int \boldsymbol{r} P(\boldsymbol{r}|\Xi, \eta) d\boldsymbol{r}. \tag{9}$$

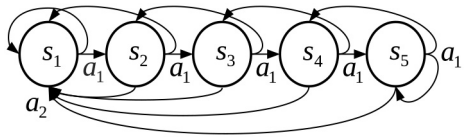


Fig. 2. 5-state chain MDP.

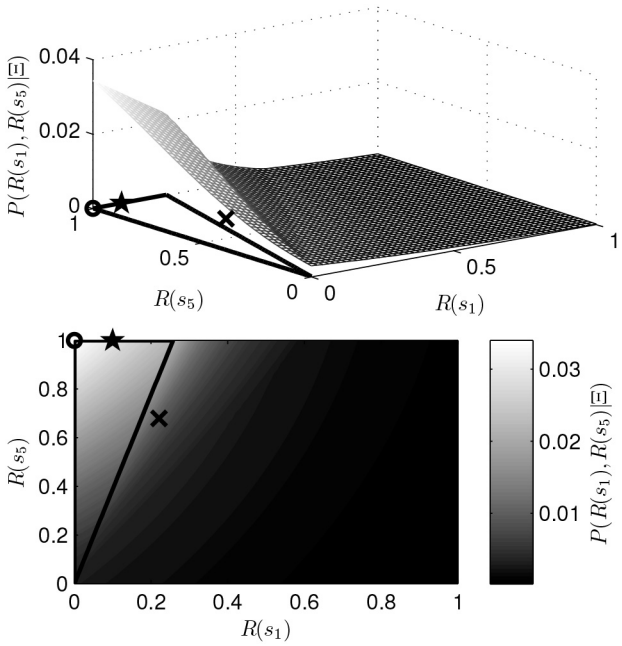


Fig. 3. Posterior distribution of the reward function in 5-state chain MDP.

A. MAP Inference in BIRL

In the general context of Bayesian inference, we can use a number of estimates for determining the reward function, such as the posterior mean, median, or maximum-a-posterior (MAP). The posterior mean is commonly used for Bayesian inference since it is the minimum mean square error estimate. However, the posterior mean can be problematic in BIRL. The posterior mean reward function minimizes the expected error over the entire space of reward functions, even including infinitely many reward functions outside the reward optimality region, which induce policies inconsistent with the behavior data. The posterior mean reward function can thus induce an optimal policy inconsistent with the data. On the other hand, the MAP is a point estimate that simply maximizes the posterior probability, hence it is not affected by the inconsistent reward functions outside the reward optimality region. Hence, it is more robust to infinitely many inconsistent reward functions. We present a simple example that compares the posterior mean and the MAP reward function estimates, defined as (9) and (11) (in Theorem 1) respectively.

Consider an MDP with five states arranged in a chain, two actions, and the discount factor 0.9. As shown in Fig. 2, we denote the leftmost state as s_1 and the rightmost state as s_5. Action a_1 moves to the state on the right with probability 0.6 and to the state on the left with probability 0.4. Action a_2 always moves to state s_1. The true reward of each state is $[0.1, 0, 0, 0, 1]$, hence the optimal policy chooses a_1 in every state. Suppose that we already know $R(s_2)$, $R(s_3)$, and $R(s_4)$ which are all 0, and estimate $R(s_1)$ and $R(s_5)$ from the behavior data Ξ which contains optimal actions for all the states. We can compute the posterior $P(R(s_1), R(s_5)|\Xi)$ using (5)–(7) under the assumption that $\mathbf{0} \leq \boldsymbol{r} \leq \mathbf{1}$ and priors $P(R(s_1))$ being $\mathcal{N}(0.1, 1)$, and $P(R(s_5))$ being $\mathcal{N}(1, 1)$. In addition, the reward optimality region can be also computed using (3).

TABLE I
IRL ALGORITHMS AND THEIR EQUIVALENT $f(\Xi; \boldsymbol{r})$ AND PRIOR FOR THE BAYESIAN FORMULATION

Previous algorithm	$f(\Xi; \boldsymbol{r})$	Prior
Ng&Russell's IRL from sampled trajectories [9]	f_V	Uniform
MMP without the loss function [10]	$(f_V)^q$	Gaussian
MWAL [18]	f_G	Uniform
Policy matching [11]	f_J	Uniform
MaxEnt [12]	f_E	Uniform

* $q \in \{1, 2\}$ is for representing L_1 or L_2 slack penalties.

Fig. 3 presents the posterior distribution of reward functions. The true reward, the MAP reward, and the posterior mean reward are marked with star, circle, and cross, respectively. The solid line is the boundary of the reward optimality region. Although the prior mean is set to the true reward, the posterior mean is outside the reward optimality region. An optimal policy for the posterior mean reward function chooses action a_2 rather than action a_1 in state s_1, while an optimal policy for the MAP reward function is identical to the true one. The situation gets worse when using the uniform prior. An optimal policy for the posterior mean reward function chooses action a_2 in states s_1 and s_2, while an optimal policy for the MAP reward function is again identical to the true one.

An additional advantage of computing the MAP reward function is in that we can view most of the previous IRL algorithms as performing the MAP estimation in BIRL. The main insight comes from the fact that these algorithms try to optimize an objective function consisting of a regularization term for the preference on the reward function and an assessment term for the compatibility of the reward function with the behavior data. We can then naturally reformulate the objective function as the posterior in BIRL by encoding the regularization into the prior and the data compatibility into the likelihood. Thus, MAP inference in BIRL subsumes different IRL approaches in the literature by generalizing the likelihood in (6) to the following:

$$P(\Xi|\boldsymbol{r}, \zeta) \propto \exp(\zeta f(\Xi; \boldsymbol{r})) \tag{10}$$

where ζ is a positive-valued parameter for scaling the likelihood and $f(\Xi; \boldsymbol{r})$ is a function encoding the data compatibility assessment used in the IRL algorithms. We thus arrive at the following result.

Theorem 1: Finding the exact solutions of the IRL algorithms listed in Table I is equivalent to computing the MAP estimates

$$\begin{aligned} \boldsymbol{r}_{\text{MAP}} &= \operatorname*{argmax}_{\boldsymbol{r}} P(\boldsymbol{r}|\Xi, \Theta) \\ &= \operatorname*{argmax}_{\boldsymbol{r}} \left[\log P(\Xi|\boldsymbol{r}, \zeta) + \log P(\boldsymbol{r}|\Theta)\right] \\ &= \operatorname*{argmax}_{\boldsymbol{r}} \left[f(\Xi; \boldsymbol{r}) + \log P(\boldsymbol{r}|\Theta)\right] \end{aligned} \tag{11}$$

where Θ is the set of parameters used in the prior and the likelihood, and $f(\Xi; \boldsymbol{r})$ is defined as follows:

$$
\begin{aligned}
f_V(\Xi; \boldsymbol{r}) &= \hat{V}^E(\boldsymbol{r}) - V^*(\boldsymbol{r}) \\
f_G(\Xi; \boldsymbol{r}) &= \min_{d \in \{1,\ldots,D\}} \left[V^{\pi(\boldsymbol{r})}(\phi_d) - \hat{V}^E(\phi_d) \right] \\
f_J(\Xi; \boldsymbol{r}) &= -\sum_{s,a} \hat{x}_E(s,a) \left(J(s,a;\boldsymbol{r}) - \hat{\pi}_E(s,a) \right)^2 \\
f_E(\Xi; \boldsymbol{r}) &= \log \mathcal{P}_{\text{MaxEnt}}(\Xi | \boldsymbol{r})
\end{aligned}
$$

where

$\pi(\boldsymbol{r})$	an optimal policy induced by $\boldsymbol{r}$;
$V^{\pi}(\phi_d)$	the value of π induced by feature function ϕ_d;
$\hat{V}^E$	the estimated value of expert's policy π_E, i.e., $\hat{V}^E = \hat{V}^{\pi_E}$;
$\hat{x}_E(s,a)$	the empirical estimate of state-action visitation counts of expert's policy π_E;
$J(s,a;\boldsymbol{r})$	a smooth mapping from $\boldsymbol{r}$ to a greedy policy;
$\mathcal{P}_{\text{MaxEnt}}$	the distribution on the behavior data satisfying the principle of maximum entropy.

In summary, the MAP inference in BIRL provides a rich framework for explaining previous non-Bayesian IRL algorithms in a unified manner, as well as encoding various types of a-priori knowledge into the prior distribution. Note that this framework can exploit insights behind other class of algorithms even though they do not have an explicit reward learning mechanism (e.g., the apprenticeship learning algorithm MWAL [18]).

B. Gradient Method for Finding the MAP Reward Function

In order to develop a gradient ascent method for the posterior optimization problem in (11), we need to show that the generalized likelihood $P(\Xi|\boldsymbol{r}, \zeta)$ defined in (10) is differentiable almost everywhere.

The likelihood is defined for measuring the compatibility of the reward function $\boldsymbol{r}$ with the behavior data Ξ. This is generally accomplished by using the optimal state-value function $\boldsymbol{v}^*$ or the optimal action-value function $\boldsymbol{q}^*$ with respect to $\boldsymbol{r}$. For example, the empirical value of Ξ can be compared with $\boldsymbol{v}^*$ [9], [10], Ξ can be directly compared to the greedy policy from $\boldsymbol{q}^*$ [11], or we can compute the probability of following the trajectories in Ξ using $\boldsymbol{q}^*$ [13]. We thus rewrite $P(\Xi|\boldsymbol{r}) = g(\Xi, \boldsymbol{v}^*(\boldsymbol{r}))$ or $g(\Xi, \boldsymbol{q}^*(\boldsymbol{r}))$ where g is differentiable with respect to $\boldsymbol{v}^*$ or $\boldsymbol{q}^*$. The remaining question is the differentiability of $\boldsymbol{v}^*$ and $\boldsymbol{q}^*$ with respect to $\boldsymbol{r}$, which we address in the following two theorems:

Theorem 2: $\boldsymbol{v}^*(\boldsymbol{r})$ and $\boldsymbol{q}^*(\boldsymbol{r})$ are convex with respect to $\boldsymbol{r}$.

Theorem 3: $\boldsymbol{v}^*(\boldsymbol{r})$ and $\boldsymbol{q}^*(\boldsymbol{r})$ are differentiable almost everywhere with respect to $\boldsymbol{r}$.

From Theorems 2 and 3, we acquire the following results: for any reward function $\boldsymbol{r}$ in the reward optimality region $C(\pi)$ with respect to π

$$
\begin{aligned}
\nabla_{\boldsymbol{r}} \boldsymbol{v}^*(\boldsymbol{r}) &= (\boldsymbol{I} - \gamma \boldsymbol{T}^{\pi})^{-1} \boldsymbol{E}^{\pi} \\
\nabla_{\boldsymbol{r}} \boldsymbol{q}^*(\boldsymbol{r}) &= (\boldsymbol{I} - \gamma \boldsymbol{T} \boldsymbol{E}^{\pi})^{-1}.
\end{aligned} \tag{13}
$$

$\nabla_{\boldsymbol{r}} \boldsymbol{v}^*(\boldsymbol{r})$ and $\nabla_{\boldsymbol{r}} \boldsymbol{q}^*(\boldsymbol{r})$ are gradients when $\boldsymbol{r}$ is strictly inside $C(\pi)$ and sub-gradients on the boundary of $C(\pi)$.

Our results in Theorems 2 and 3 are related to the previous work on gradient methods for IRL. Neu and Szepesvári [11] showed that $\boldsymbol{q}^*(\boldsymbol{r})$ is Lipschitz continuous, and it is Fréchet differentiable except on a set of measure zero (almost everywhere) by Rademacher's theorem. We have obtained the same result based on the reward optimality region, and additionally identified the condition for which $\boldsymbol{v}^*(\boldsymbol{r})$ and $\boldsymbol{q}^*(\boldsymbol{r})$ are non-differentiable. Ratliff *et al.* [10] used a sub-gradient of their objective function because it involves differentiating $\boldsymbol{v}^*(\boldsymbol{r})$. Computing the sub-gradient of their objective function yields an identical result using (13).

With a differentiable prior, we can compute the (sub) gradient of the posterior using (13) and the chain rule. If the prior and g are convex, then the posterior will be convex and we will find the MAP reward function. Otherwise, we will obtain a locally optimal solution as in Neu and Szepesvári [11].

As an example, we show how we can calculate the gradient of the posterior in BIRL when the likelihood in (6) and the independent normal prior in (8) are used. The log unnormalized posterior is computed as

$$
\begin{aligned}
&\log P(\boldsymbol{r}|\Xi, \eta, \mu, \sigma) \\
&\quad \propto \log P(\Xi|\boldsymbol{r}, \eta) + \log P(\boldsymbol{r}|\mu, \sigma) \\
&\quad = \sum_{m,h} \log \psi(a_{m,h}|s_{m,h}; \boldsymbol{r}, \eta) + \sum_{s,a} \log \mathcal{N}(r_{s,a}|\mu, \sigma).
\end{aligned}
$$

We then use (13) to obtain the gradient which is shown in Table II. We also note that, when using a normal prior with zero mean, maximizing the posterior is equivalent to optimizing the compatibility of the reward function with L_2 regularization.

IV. Hierarchical BIRL

In this section, we present a hierarchical Bayesian approach to IRL (HBIRL), which extends BIRL in two ways: imposing a prior on the confidence parameter and a hyper-prior on the prior of the reward function.[3]

Although most of the previous IRL algorithms assume that the expert behaves optimally, she can choose sub-optimal actions in reality: since human beings (and animals in general) have limited time, knowledge, and computational capabilities, they cannot consider all the possible future consequences of actions and may behave sub-optimally. This is referred to as bounded rationality [15]–[17]. They also occasionally fail to choose the best action simply by mistake. There is a wealth of literature on human sub-optimal behavior in psychology and economics [24], [25].

BIRL implicitly handled the sub-optimal actions by using the softmax distribution in the likelihood (6), which yields the probability of selecting an optimal action. In the softmax distribution, the parameter η represents the confidence of choosing optimal actions. The probability mass is concentrated on the optimal actions with the maximum Q^*'s for large η, and is spread over all actions with small η. Hence, it is convenient

[3]Our hierarchical model currently assumes a specific prior for the reward function so that we can choose a conjugate hyper-prior and obtain analytical formulas of marginalized gradients.

TABLE II
GRADIENT OF THE POSTERIOR IN THE BIRL MODEL

$$\nabla_{\boldsymbol{r}} \log P(\boldsymbol{r}|\Xi, \eta, \mu, \sigma) \propto \sum_{m,h} \eta \left[\nabla_{\boldsymbol{r}} q^*_{s_{m,h}, a_{m,h}}(\boldsymbol{r}) - \sum_a \psi(s_{m,h}, a; \boldsymbol{r}, \eta) \nabla_{\boldsymbol{r}} q^*_{s_{m,h}, a}(\boldsymbol{r}) \right] - \frac{1}{2\sigma^2} \sum_{s,a} (r_{s,a} - \mu)^2 \tag{12}$$

to use the softmax distribution to assign a nonzero probability to the expert's sub-optimal action selection. However, it is nontrivial to manually choose an appropriate value for η that adequately represents the degree of the expert's sub-optimality.

HBIRL imposes a prior on η to explicitly estimate the expert's sub-optimality from the behavior data. Specifically, we used a gamma distribution as the prior on η since it is sufficiently flexible to exhibit various shapes of distributions on a nonnegative real-valued random variable using only two parameters

$$P(\eta|\boldsymbol{\tau}) = \mathcal{G}(\eta; \tau_1, \tau_2) \tag{14}$$

where τ_1 is the shape parameter, τ_2 is the inverse scale parameter, and $\boldsymbol{\tau} = [\tau_1, \tau_2]$.[4]

Revisiting BIRL, we remind that it is capable of incorporating a-priori domain knowledge into the model by choosing a suitable prior distribution on reward functions. However, a number of issues still remain to be resolved. For example, it is often nontrivial to encode the domain knowledge into the prior. When specifying the prior, we need to choose a specific distribution along with its parameter values. Even after all the effort, we may choose the parameter values far from the true ones as well as the distribution itself. This is particularly harmful when the data is sparse. It is thus desirable to learn the prior distribution from the behavior data for robust IRL.

HBIRL addresses this issue by defining a hyper-prior on the prior of reward function. We first assume that the rewards are independently and normally distributed since the normal distribution is widely used to model a real-valued random variable when no information other than its mean and variance are available [26]. The prior of the reward function is thus defined as

$$P(\boldsymbol{r}|\boldsymbol{\mu}, \boldsymbol{\lambda}) = \prod_{d=1}^{D} P(r_d|\mu_d, \lambda_d) = \prod_{d=1}^{D} \mathcal{N}\left(r_d; \mu_d, \lambda_d^{-1}\right)$$

where $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_D]^\intercal$ and $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_D]^\intercal$, where μ_d and λ_d are the mean and precision of r_d, respectively. We then use the normal-gamma distribution as the hyper-prior on the prior of reward function, since it is the conjugate prior for the mean and precision of the normal distribution

$$P(\mu_d, \lambda_d|\beta, \boldsymbol{\gamma}) = \mathcal{N}\left(\mu_d; 0, (\beta\lambda_d)^{-1}\right) \mathcal{G}(\lambda_d; \gamma_1, \gamma_2)$$

where β, γ_1, and γ_2 are the parameters for the normal-gamma distribution[5] and $\boldsymbol{\gamma} = [\gamma_1, \gamma_2]$. We thus have

$$P(r_d|\beta, \boldsymbol{\gamma}) = \int P(r_d|\mu_d, \lambda_d) \, \mathrm{d}P(\mu_d, \lambda_d|\beta, \boldsymbol{\gamma}). \tag{15}$$

[4]We used a vague prior with $\tau_1 = \tau_2 = 1$ throughout the experiments.

[5]We again used a vague prior by specifying $\beta = \gamma_1 = \gamma_2 = 1$ throughout the experiments.

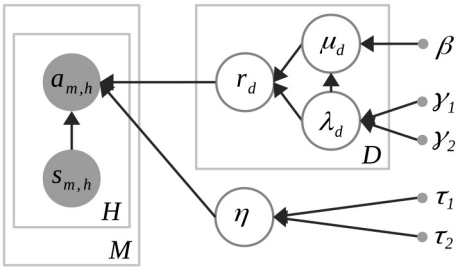


Fig. 4. Graphical representation of the HBIRL model.

The prior on the confidence parameter η and the reward function $\boldsymbol{r}$ is now ready to be defined, combining (14) and (15) as follows:

$$\begin{aligned} P(\boldsymbol{r}, \eta|\beta, \boldsymbol{\gamma}, \boldsymbol{\tau}) &= P(\eta|\boldsymbol{\tau}) P(\boldsymbol{r}|\beta, \boldsymbol{\gamma}) \\ &= P(\eta|\boldsymbol{\tau}) \prod_{d=1}^{D} P(r_d|\beta, \boldsymbol{\gamma}). \end{aligned}$$

The posterior on $\boldsymbol{r}$ and η is then formulated by combining the likelihood in (6) with the prior

$$P(\boldsymbol{r}, \eta|\Xi, \beta, \boldsymbol{\gamma}, \boldsymbol{\tau}) \propto P(\Xi|\boldsymbol{r}, \eta) P(\boldsymbol{r}, \eta|\beta, \boldsymbol{\gamma}, \boldsymbol{\tau}). \tag{16}$$

Fig. 4 depicts a graphical representation of the HBIRL model.

We then find the MAP estimate of reward function $\boldsymbol{r}_{\mathrm{MAP}}$ and confidence parameter η_{MAP} by reformulating the IRL problem as an optimization problem with the objective being maximization of the (log unnormalized) posterior

$$\langle \boldsymbol{r}_{\mathrm{MAP}}, \eta_{\mathrm{MAP}} \rangle = \underset{\boldsymbol{r}, \eta}{\operatorname{argmax}} \log P(\boldsymbol{r}, \eta|\Xi, \Theta)$$

where $\Theta = \{\beta, \boldsymbol{\gamma}, \boldsymbol{\tau}\}$. The log unnormalized posterior can be computed as

$$\begin{aligned} \log P(\boldsymbol{r}, \eta|\Xi, \Theta) &\propto \log P(\Xi|\boldsymbol{r}, \eta) + \log P(\boldsymbol{r}, \eta|\beta, \boldsymbol{\gamma}, \boldsymbol{\tau}) \\ &= \sum_{m,h} \log \psi(a_{m,h}|s_{m,h}; \boldsymbol{r}, \eta) \\ &\quad + \log P(\eta|\boldsymbol{\tau}) + \sum_{d} \log P(r_d|\beta, \boldsymbol{\gamma}). \end{aligned}$$

Since the gradient of $\boldsymbol{q}^*(\boldsymbol{r})$ with respect to $\boldsymbol{r}$ can be computed using Theorem 3, we can compute the gradient of the posterior with respect to $\boldsymbol{r}$ and η as shown in Table III. We then find the approximate MAP estimate of $\boldsymbol{r}_{\mathrm{MAP}}$ and η_{MAP} by using a gradient ascent method, as shown in Table IV.

V. EXPERIMENT RESULTS

We empirically compared the performance of our gradient methods in BIRL and HBIRL to those of other IRL algorithms in the literature: Abbeel and Ng's projection algorithm [14],

TABLE III
GRADIENT OF THE POSTERIOR FROM THE HBIRL MODEL

$$\nabla_{\boldsymbol{r}} \log P(\boldsymbol{r}, \eta | \Xi, \beta, \boldsymbol{\gamma}, \boldsymbol{\tau}) \propto \sum_{m,h} \eta \left[\nabla_{\boldsymbol{r}} q^*_{s_{m,h}, a_{m,h}}(\boldsymbol{r}) - \sum_a \psi(s_{m,h}, a; \boldsymbol{r}, \eta) \nabla_{\boldsymbol{r}} q^*_{s_{m,h}, a}(\boldsymbol{r}) \right] - 2\beta(\gamma_1 + \frac{1}{2}) \sum_{d=1}^{D} \frac{r_d}{2(\beta+1)\gamma_2 + \beta r_d^2}$$

$$\nabla_{\eta} \log P(\boldsymbol{r}, \eta | \Xi, \beta, \boldsymbol{\gamma}, \boldsymbol{\tau}) \propto \sum_{m,h} \left[q^*_{s_{m,h}, a_{m,h}}(\boldsymbol{r}) - \sum_a \psi(s_{m,h}, a; \boldsymbol{r}, \eta) q^*_{s_{m,h}, a}(\boldsymbol{r}) \right] + \frac{\tau_1 - 1}{\eta} - \tau_2$$

TABLE IV
GRADIENT ASCENT FOR MAP INFERENCE IN HBIRL

Input: MDP$\backslash R$, behavior data Ξ, step-size sequence $\{\delta_t\}$
Output: $\langle \boldsymbol{r}_{\text{MAP}}, \eta_{\text{MAP}} \rangle$

```
Initialize r and η
while r and η not converged do
    π ← solve_mdp(r)
    % use the formula for gradients in Table III
    r ← r + δ_t ∇_r log P(r, η|Ξ, β, γ, τ)
    η ← η + δ_t ∇_η log P(r, η|Ξ, β, γ, τ)
end while
⟨r_MAP, η_MAP⟩ ← ⟨r, η⟩
```

Maximum Margin Planning (MMP) [10], Maximum Entropy (MaxEnt) IRL [12], and an MCMC method in BIRL [13]. However, we omit the results of Abbeel and Ng's projection and MMP algorithms since they performed significantly worse than the other algorithms. The methods in BIRL used the standard normal distribution as the prior and we set the confidence parameter $\eta = 1$. We stopped the gradient methods when the parameter values converged and the MCMC methods when the number of iterations reached 20000. All the algorithms were implemented in MATLAB, and the gradient ascent method used `fmincon` which automatically selects the stepsize δ_t.

To evaluate the performance of the IRL algorithms, we could directly measure the distance between the expert's reward function and the one found by the algorithms. However, this can yield meaningless results since the reward represents the relative importance of executing an action in a state. Two reward functions having a small difference may yield completely different policies, and two reward functions having a large difference may yield an identical policy. For example, the degenerate reward function $\boldsymbol{r} = c\mathbf{1}$ where $c \in [-R_{\max}, R_{\max}]$ makes any policy optimal. Hence, we evaluated the performance of the algorithms by comparing the optimal policies induced by the expert's and the learned reward functions. We define the measure as the expected value difference (EVD)

$$\frac{1}{|\mathcal{S}|} \left|\left| \boldsymbol{v}^{\pi(\boldsymbol{r}_E)}(\boldsymbol{r}_E) - \boldsymbol{v}^{\pi(\boldsymbol{r}_L)}(\boldsymbol{r}_E) \right|\right|_1$$

where $\boldsymbol{r}_E$ is the expert's reward functions (i.e., the ground truth), $\boldsymbol{r}_L$ is the learned reward function, and $\pi(\boldsymbol{r})$ is an optimal policy induced by reward function $\boldsymbol{r}$. The EVD measures the loss in the optimality incurred by using the policy from the learned reward function instead of the expert's reward function. We also define the miss prediction rate of $\pi(\boldsymbol{r}_L)$ in the expert's optimal behavior Ξ' as

$$\frac{1}{M'H'} \sum_{m=1}^{M'} \sum_{h=1}^{H'} \mathbf{1}\left(a_{m,h} \neq \pi\left(s_{m,h}; \boldsymbol{r}_L\right)\right)$$

where $\mathbf{1}(x)$ is the indicator function (1 if x is true and 0 otherwise), and $\Xi' = \{\xi'_1, \ldots, \xi'_{M'}\}$ is a held-out test dataset of M' trajectories of length H' generated from $\pi(\boldsymbol{r}_E)$. The miss prediction rate measures the loss in the incorrect action prediction via the zero-one loss function.

A. Gridworld Problem

We performed experiments on the gridworld problem, a synthetic problem domain widely used in the IRL literature [11], [14], [27], [28]. In this problem, the agent can move in one of the four directions or stay in the current location on a $n \times n$ grid. Each action fails with a probability of 0.3 and moves the agent in a random direction. The grid is partitioned into nonoverlapping regions of size 2×2, resulting in $(n/2)^2$ regions. The feature functions are defined as the binary indicator functions for each region. 30% of the regions were randomly sampled and the rewards for those regions were sampled i.i.d from the standard normal distribution.

The first set of experiments concerns with learning from optimal experts: we prepared training data consisting of $10n$ trajectories with $2n$ time steps, which was collected from the simulated runs of optimal policies. Fig. 5 shows the average performance including the standard error on 20 experimental runs. The panels in the top row represent the performance by varying the number of trajectories in the 24×24 gridworld, and the panels in the bottom row represent the performance in varying sizes of gridworlds. We can make the following observations from the results.

1) *Top Left and Top Right:* As more trajectories are gathered for the training data, we naturally expect that the algorithms will produce better results, i.e., lower EVDs and miss prediction rates. Note that the EVD curve is more stable than the miss prediction rate curve for every algorithm.
2) *Bottom Left and Bottom Right*: As the state space is enlarged, we naturally expect that the algorithms will produce worse results due to a number of problems including data sparsity. Again, note that the EVD curve is more stable than the miss prediction rate curve for every algorithm. These results show that the EVD is a better choice for measuring performance than the miss

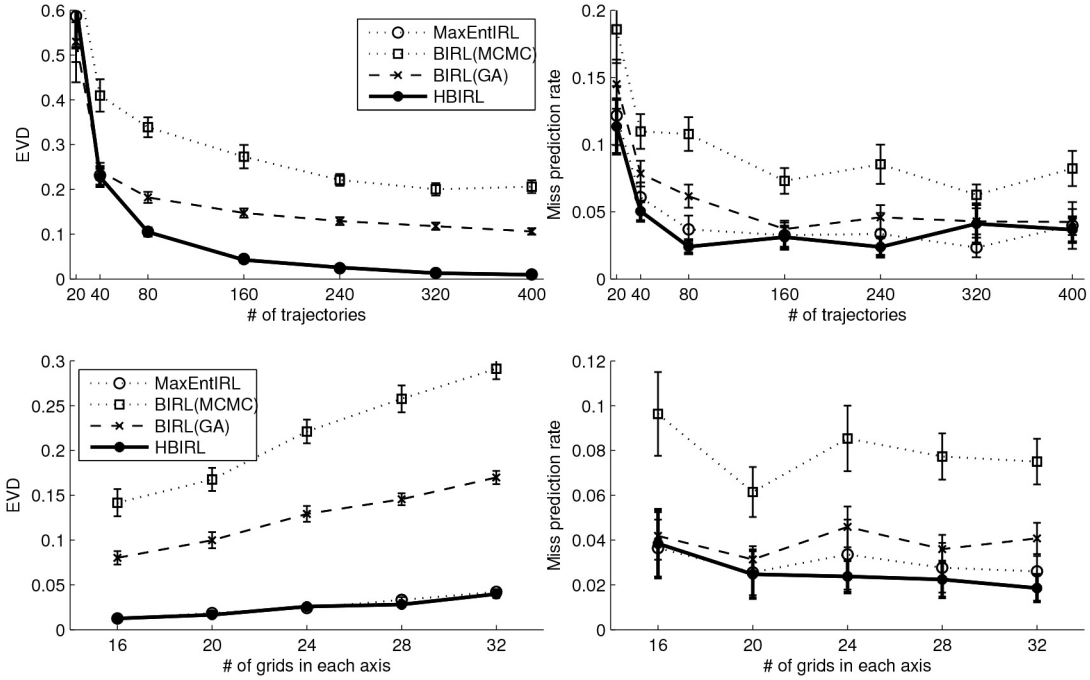


Fig. 5. Results with optimal experts in the gridworld problems.

prediction rate, since the latter simply counts the number of incorrect actions whereas the former additionally measures how bad the incorrect actions are.

3) *Top Left and Bottom Left:* Two gradient ascent methods (BIRL with GA and HBIRL) performed better than BIRL with MCMC, which demonstrates that MAP estimate is more effective than the posterior mean estimate.
4) *Top Left and Bottom Left:* HBIRL significantly outperformed BIRL, which was achieved by adapting the confidence parameter and the reward function prior to the data (e.g., η was properly estimated to be much higher than its default value of 1 since the data does not contain any sub-optimal action).
5) *Top Left and Bottom Left:* HBIRL achieved almost the same level of performance as that of MaxEnt IRL, which is one of the best performing algorithms in the IRL literature.

In the second set of experiments, we measured the performance of the IRL algorithms on sub-optimal behavior data. We prepared training data consisting of 240 trajectories with 48 time steps in the 24×24 gridworld using two types of sub-optimal experts using the models of bounded rationality. First, the ϵ-optimized policy model $\tilde{\pi}_\epsilon$ samples actions uniformly whose values are within ϵ of the optimum

$$\tilde{\pi}_\epsilon(s,a) = \begin{cases} \frac{1}{|\mathcal{A}^*_\epsilon|} & \text{if } a \in \mathcal{A}^*_\epsilon \\ 0 & \text{otherwise} \end{cases}$$

where $\mathcal{A}^*_\epsilon = \{a \in \mathcal{A} | Q^*(s,a) \geq \max_{a' \in \mathcal{A}} Q^*(s,a') - \epsilon\}$. This is a model used in economics for decision makers who behave according to some heuristic capable of being reasonably close to the optimum [29]. Second, the h-step look-ahead policy model $\tilde{\pi}_h$ reflects that the decision maker cannot plan optimally for the full horizon because of limited computational capabilities. This models the human decision restricted by a finite amount of resource for reasoning [17]. Fig. 6 shows the average performance with the standard error over 20 training data for various values of ϵ and h. We can make the following observations from the results on sub-optimal behavior data.

1) *Top Left:* The EVD of BIRL under ϵ-optimized policy shows a U-shaped curve, which is a naturally expected result since BIRL achieves minimum EVD when ϵ is set to the equivalent value of the default setting $\eta = 1$.
2) *Top Right:* On the other hand, the trend in the miss prediction rate is less evident. The reason for the apparent inconsistency once again comes from the crude performance measurement made by miss prediction rate.
3) *Top Left:* All the algorithms achieved lower EVDs compared to the policy $\tilde{\pi}_\epsilon$ (Expert) when ϵ is large. This result shows that all the algorithms have the basic capability of handling sub-optimal behavior data for inferring the true optimal policy.
4) *Top Left:* BIRL is outperformed by MaxEnt IRL for small ϵ, but it performs better for large ϵ. This is due to the fact that the default value for η used in BIRL is not suited for small ϵ, but the softmax distribution used in BIRL is a more robust model for the data generated from ϵ-optimized policies with large ϵ.
5) *Top Left:* HBIRL performed the best among the three algorithms for all ϵ. This result shows that HBIRL successfully adapted the confidence parameter η to reflect the sub-optimality in the behavior data.
6) *Top Right:* Once again, the miss prediction rate yields less reliable results compared to the EVD.
7) *Bottom Left:* When the h-step look-ahead policy $\tilde{\pi}_h$ is used to generate the behavior data, all the algorithms tend to produce better results as the horizon h is increased since the policy becomes closer to the optimal policy.

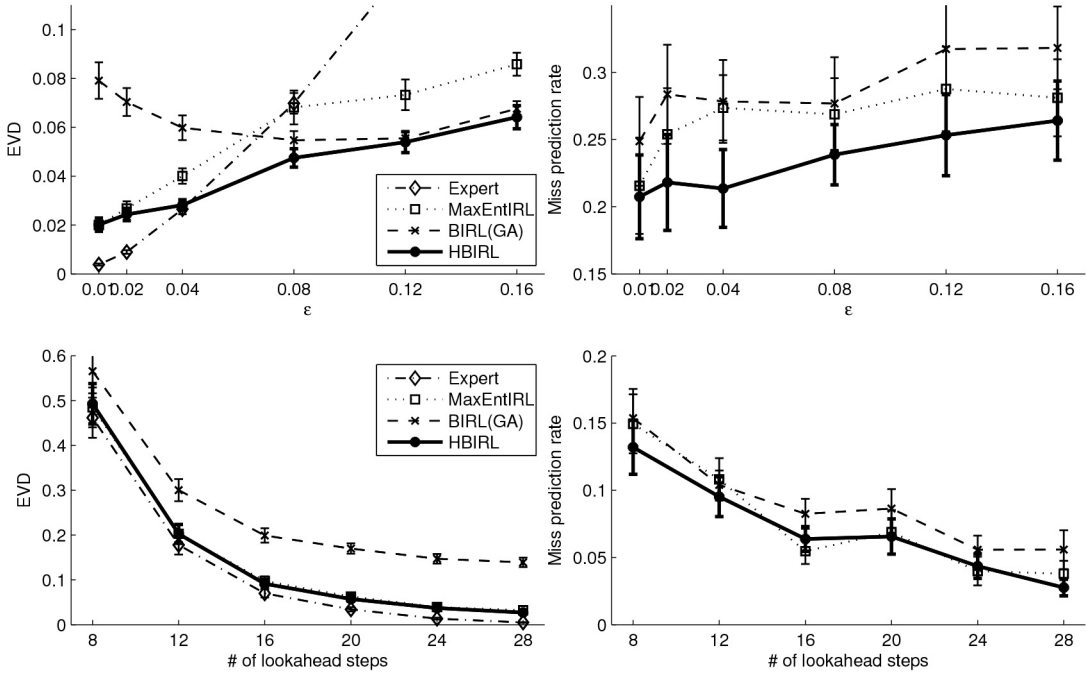


Fig. 6. Results with sub-optimal experts using ϵ-optimized policies (top row) and h-step look-ahead policies (bottom row) in the gridworld problems.

8) *Bottom Right:* we observe a similar trend in the miss prediction rate, although less stable than in the EVD case.
9) *Bottom Left:* The policies found by the algorithms are no better than $\tilde{\pi}_h$ (Expert), in contrast to the results with $\tilde{\pi}_\epsilon$. This is because $\tilde{\pi}_h$, which always executes optimal actions up to horizon h and then behaves randomly, cannot be properly represented by the IRL algorithms which assume a constant level of sub-optimality for an infinite horizon.
10) *Bottom Left:* HBIRL performs significantly better than BIRL, and achieves almost the same level of performance as that of MaxEnt IRL, which is one of the best performing algorithms in the IRL literature.

In summary, HBIRL significantly outperformed BIRL, and performed on par with MaxEnt IRL, which employs maximum entropy principle to deal with action uncertainty. Furthermore, HBIRL outperformed MaxEnt IRL when the behavior data was gathered from ϵ-optimized policy, in which case the action probability model (i.e., softmax distribution) provides a suitable approximation of the sub-optimal behavior.

As a final note, it may be seen that we should always prefer the EVD to the miss prediction rate, since the former measures the performance in a finer detail. Although this is true, we shall remind that the EVD uses the MDP model to compute the value function. When we construct the MDP model from data (as done in the next section), the inevitable modeling bias can render EVD misleading. Hence, the miss prediction rate can be preferable to the EVD when we want to measure the performance purely based on the data. However, as shown in the sub-figures on the right in Fig. 6, it is not evident that HBIRL performs significantly better than BIRL or MaxEntIRL due to the increase in standard error.

B. Taxi Driver Behavior Prediction

In this section, we infer taxi drivers' route preference using the GPS trace data collected in the San Francisco area [30]. We retrieved the road network information from the OPENSTREETMAP,[6] and represented it as a graph consisting of 13723 intersections and 20518 road segments. As in Ziebart *et al.* [4], we assumed that taxi drivers try to reach a destination in a trip by taking a sequence of road segments according to their preferences. Among the traces of 500 taxis in the original data, we selected 1861 trips from 20 taxis, depicted in Fig. 7. The total distance of the trips was 4335 miles. We then segmented the GPS traces and mapped them to the road segments using a hidden Markov model [31].

We used the goal-oriented MDP $\mathcal{M}_g$ [32] to model the problem, where the objective of the decision maker is to reach the goal state g while minimizing the expected total cost (or equivalently, maximizing the expected total reward which is negative of the cost). The rewards are assumed to be negative everywhere except at the goal state, which is an absorbing state with a zero reward. The states in $\mathcal{M}_g$ are road segments (20518 states) and the state transitions correspond to taking one of the road segments at intersections. For each trip, the road segment corresponding to the destination was designated as the goal state g (1861 goal states).

We prepared two kinds of features. The state-dependent features represent the properties of the road segments, such as the type, the speed limit, the number of lanes, one-way, tunnel, and bridge, by the length of road segments.[7] The action-dependent features represent the angle of the turn by

[6]http://www.openstreetmap.org

[7]Type $\in$ {highway, primary street, secondary street, living street}, speed limit $\in$ {below 20 mph, 20-30 mph, 30-40 mph, above 40 mph}, # lanes $\in \{1, 2, 3+\}$. These features were obtained using the OPENSTREETMAP.

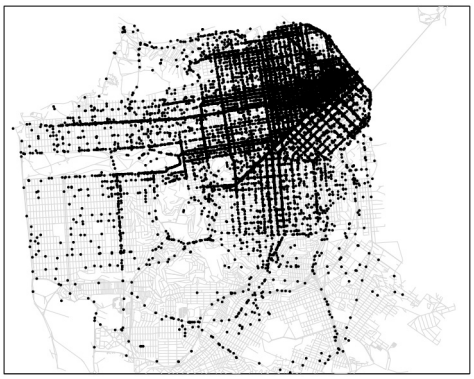


Fig. 7. Road network (gray lines) and chosen GPS traces (black dots) in San Francisco.

TABLE V
DRIVER BEHAVIOR PREDICTION RESULTS

	Turn prediction (%)	Route prediction (%)
Random policy	43.28 (± 0.15)	13.24 (± 0.04)
Shortest path	88.28 (± 0.09)	43.88 (± 0.29)
MDP ($\boldsymbol{r}_H$)	91.89 (± 0.08)	51.89 (± 0.16)
MaxEntIRL	89.12 (± 0.06)	44.19 (± 0.24)
BIRL	92.04 (± 0.12)	54.57 (± 0.46)
HBIRL	92.12 (± 0.07)	55.48 (± 0.27)

binary indicator functions.[8] Given the destination g, the reward function was calculated as $\boldsymbol{r}_g = \boldsymbol{F}_g \cdot \boldsymbol{w}$ where $\boldsymbol{F}_g$ denotes an $|\mathcal{S}||\mathcal{A}| \times D$ feature matrix and $\boldsymbol{w}$ denotes a D-dimensional reward weight vector. $\boldsymbol{F}_g$ is defined as the set of predefined features $\{\phi_1, \ldots, \phi_D\}$ conditioned on the destination g as

$$F_g(\langle g, a\rangle, d) = 0$$
$$F_g(\langle s, a\rangle, d) = \phi_d(s, a), \text{ for } \forall s \in \mathcal{S}\backslash\{g\}.$$

The reward weight vector is constrained to be negative ($w_d < 0$) to be consistent with the definition of goal-oriented MDPs. This leads to a slight modification to the likelihood so that

$$P(\Xi|\boldsymbol{r}, \eta) = \prod_m P(\xi_m|\boldsymbol{r}_{g_m}, \eta; \mathcal{M}_{g_m})$$

where g_m is the destination in the trajectory ξ_m. Note that, although we have different MDPs $\mathcal{M}_{g_m}$ for each trajectory due to different destination, they are assumed to share the same reward weight vector $\boldsymbol{w}$.

We evaluated the prediction accuracies in terms of the turn and the route predictions. The turn prediction accuracy measures the ratio of the correct actions taken at the intersections in the route. The route prediction accuracy measures the ratio of the total distance of the correct road segments in the route.

We compared the BIRL and HBIRL to four other methods including MaxEntIRL. We used a random policy as for the first baseline and the shortest path to the destination as for the second baseline. The third baseline method was the optimal policy from an MDP with heuristically chosen reward $\boldsymbol{r}_H$. A simple heuristic would be counting the feature visitations in the training data, and set the reward accordingly. Since we restrict the reward to be negative, we used the normalized counts of feature visitations offset by the maximum count value.

We evaluated the algorithms via fourfold cross validation. Table V shows the average prediction accuracies and their standard errors. Since we modeled the problem as the deterministic MDP and used a small number of features, the MDP optimal policy from the heuristically chosen reward $\boldsymbol{r}_H$ recovered the drivers' preference fairly well, even performing better than MaxEntIRL. In contrast, BIRL and HBIRL outperformed other methods, HBIRL achieving the best performance.

[8]Turn angle ∈ {hard left, soft left, straight, soft right, hard right, u-turn}.

VI. CONCLUSION

In this paper, we presented HBIRL, a hierarchical Beysian model of IRL. HBIRL extends BIRL by imposing a prior on the parameter of policies to infer the expert's sub-optimality, and a hyper-prior on the reward function to mitigate the difficulty of specifying an appropriate prior for the reward function. We also provided a gradient method to find the MAP estimate of the reward function of the hierarchical model, based on the derivation of analytical formulas for computing the gradients.

Through experiments on synthetic datasets, we showed the effectiveness of the MAP estimation over the posterior mean estimation, and demonstrated the robustness of HBIRL to the expert's sub-optimal behavior. Additionally, we demonstrated that this approach can be used for real-world data, predicting taxi drivers' route selection using a real GPS trace dataset, and showed that the proposed methods achieve higher accuracies than previous methods.

There are a number of promising directions for future work on extending our approach. First, we presented the analytical formulas of gradients for a specific choice of reward function prior and likelihood in BIRL. Extending our derivation results to other priors and likelihoods with an appropriate choice of hyper-priors will strengthen the applicability of HBIRL. Second, although we only covered reward function learning, extending HBIRL to policy learning can provide a new set of tools for apprenticeship learning. Third, it would be interesting to extend the approach to hierarchical representation of policies motivated by hierarchical reinforcement learning [33]–[35].

APPENDIX

A. Proof of Corollary 1

Proof: Policy π is optimal

$$\begin{aligned}
&\Leftrightarrow \boldsymbol{q}^{\pi(\boldsymbol{r}),a} \le \boldsymbol{v}^{\pi}(\boldsymbol{r}) \\
&\Leftrightarrow \boldsymbol{r}^a + \gamma \boldsymbol{T}^a \boldsymbol{v}^{\pi}(\boldsymbol{r}) \le \boldsymbol{r}^{\pi} + \gamma \boldsymbol{T}^{\pi} \boldsymbol{v}^{\pi}(\boldsymbol{r}) \\
&\Leftrightarrow \boldsymbol{r}^a + \gamma \boldsymbol{T}^a (\boldsymbol{I} - \gamma \boldsymbol{T}^{\pi})^{-1} \boldsymbol{r}^{\pi} \\
&\quad \le \boldsymbol{r}^{\pi} + \gamma \boldsymbol{T}^{\pi} (\boldsymbol{I} - \gamma \boldsymbol{T}^{\pi})^{-1} \boldsymbol{r}^{\pi} \\
&\Leftrightarrow \boldsymbol{r}^a - (\boldsymbol{I} - \gamma \boldsymbol{T}^a)(\boldsymbol{I} - \gamma \boldsymbol{T}^{\pi})^{-1} \boldsymbol{r}^{\pi} \\
&\quad \le \boldsymbol{r}^{\pi} - (\boldsymbol{I} - \gamma \boldsymbol{T}^{\pi})(\boldsymbol{I} - \gamma \boldsymbol{T}^{\pi})^{-1} \boldsymbol{r}^{\pi} \\
&\Leftrightarrow \boldsymbol{r}^a - (\boldsymbol{I} - \gamma \boldsymbol{T}^a)(\boldsymbol{I} - \gamma \boldsymbol{T}^{\pi})^{-1} \boldsymbol{E}^{\pi} \boldsymbol{r} \le \boldsymbol{0}. \qquad (17)
\end{aligned}$$

The third equivalence comes from $\boldsymbol{v}^{\pi}(\boldsymbol{r}) = (\boldsymbol{I} - \gamma \boldsymbol{T}^{\pi})^{-1} \boldsymbol{r}^{\pi}$. The fifth equivalence holds because $\boldsymbol{r}^{\pi} - (\boldsymbol{I} - \gamma \boldsymbol{T}^{\pi})$

$(\boldsymbol{I} - \gamma \boldsymbol{T}^{\pi})^{-1}\boldsymbol{r}^{\pi} = \boldsymbol{0}$ and $\boldsymbol{r}^{\pi} = \boldsymbol{E}^{\pi}\boldsymbol{r}$. Stacking (17) for all $a \in \mathcal{A}$, we obtain (3). ■

B. Proof of Theorem 1

We prove Theorem 1 by the following lemmas.

Lemma 1: The reward function sought by Ng and Russell's IRL algorithm from sampled trajectories [9] is equivalent to the MAP estimate with the uniform prior and the likelihood using $f_V(\Xi; \boldsymbol{r}) = \hat{V}^E(\boldsymbol{r}) - V^*(\boldsymbol{r})$.

Proof: This IRL algorithm seeks the reward function defined by

$$\boldsymbol{r}_{\text{N\&R}} = \operatorname*{argmax}_{\boldsymbol{r}} \left[\hat{V}^E(\boldsymbol{r}) - V^*(\boldsymbol{r})\right].$$

The MAP estimate with the uniform prior and the likelihood using f_V is computed as

$$\begin{aligned}\boldsymbol{r}_{\text{MAP}} &= \operatorname*{argmax}_{\boldsymbol{r}} P(\boldsymbol{r}|\Xi) = \operatorname*{argmax}_{\boldsymbol{r}} \log P(\boldsymbol{r}|\Xi) \\ &= \operatorname*{argmax}_{\boldsymbol{r}} \left[\log P(\Xi|\boldsymbol{r}) + \log P(\boldsymbol{r})\right] \\ &= \operatorname*{argmax}_{\boldsymbol{r}} f_V(\Xi; \boldsymbol{r}) \\ &= \operatorname*{argmax}_{\boldsymbol{r}} \left[\hat{V}^E(\boldsymbol{r}) - V^*(\boldsymbol{r})\right].\end{aligned}$$

The MAP estimate is thus equivalent to $\boldsymbol{r}_{\text{N\&R}}$. ■

Lemma 2: The reward function sought by the MMP algorithm [10] without the loss function is equivalent to the MAP estimate with a Gaussian prior and the likelihood using $(f_V)^q$ where $q \in \{1, 2\}$.

Proof: Without the loss function, the MMP algorithm seeks the reward function defined by

$$\boldsymbol{r}_{\text{MMP}} = \operatorname*{argmin}_{\boldsymbol{r}} \left[\left(V^*(\boldsymbol{r}) - \hat{V}^E(\boldsymbol{r})\right)^q + \frac{\lambda}{2}||\boldsymbol{r}||_2^2\right]$$

where $q \in \{1, 2\}$ denotes L_1 or L_2 slack penalties. The MAP estimate with a Gaussian prior $\mathcal{N}(0, \sigma^2)$ and the likelihood using $(f_V)^q$ is computed as

$$\begin{aligned}\boldsymbol{r}_{\text{MAP}} &= \operatorname*{argmax}_{\boldsymbol{r}} P(\boldsymbol{r}|\Xi) = \operatorname*{argmax}_{\boldsymbol{r}} \left[\log P(\Xi|\boldsymbol{r}) + \log P(\boldsymbol{r})\right] \\ &= \operatorname*{argmax}_{\boldsymbol{r}} \left[\zeta\,(f_V(\Xi; \boldsymbol{r}))^q - \frac{1}{2\sigma^2}\sum_{s,a}\boldsymbol{r}(s, a)^2\right] \\ &= \operatorname*{argmax}_{\boldsymbol{r}} \left[(f_V(\Xi; \boldsymbol{r}))^q - \frac{1}{2\zeta\sigma^2}||\boldsymbol{r}||_2^2\right] \\ &= \operatorname*{argmin}_{\boldsymbol{r}} \left[\left(V^*(\boldsymbol{r}) - \hat{V}^E(\boldsymbol{r})\right)^q + \frac{1}{2\zeta\sigma^2}||\boldsymbol{r}||_2^2\right].\end{aligned}$$

If we set $\lambda = 1/(\zeta\sigma^2)$, the MAP estimate is equivalent to R_{MMP}. ■

Lemma 3: When the reward function is linearly parameterized using the reward weight vector $\boldsymbol{r}$ such that $\sum_d r_d = 1$ and $r_d \geq 0$, the policy sought by the MWAL algorithm [18] is equivalent to an optimal policy on the reward function which is the MAP estimate with the uniform prior and the likelihood using $f_G(\Xi; \boldsymbol{r}) = \min_d [V^{\pi(\boldsymbol{r})}(\phi_d) - \hat{V}^E(\phi_d)]$ where $\pi(\boldsymbol{r})$ is an optimal policy induced by $\boldsymbol{r}$.

Proof: The MWAL algorithm seeks the policy π_{MWAL} defined by

$$\pi_{\text{MWAL}} = \operatorname*{argmax}_{\pi} \min_d \left[V^{\pi}(\phi_d) - \hat{V}^E(\phi_d)\right]$$

with an implicitly computed reward function $\boldsymbol{r}_{\text{MWAL}}$ that induces π_{MWAL} as an optimal policy. Hence, we can rewrite $\pi_{\text{MWAL}} = \pi(\boldsymbol{r}_{\text{MWAL}})$ where

$$\boldsymbol{r}_{\text{MWAL}} = \operatorname*{argmax}_{\boldsymbol{r}} \min_d \left[V^{\pi(\boldsymbol{r})}(\phi_d) - \hat{V}^E(\phi_d)\right].$$

The MAP estimate of the reward function with the uniform prior and the likelihood using f_G is computed as

$$\begin{aligned}\boldsymbol{r}_{\text{MAP}} &= \operatorname*{argmax}_{\boldsymbol{r}} P(\boldsymbol{r}|\Xi) \\ &= \operatorname*{argmax}_{\boldsymbol{r}} f_G(\Xi; \boldsymbol{r}) \\ &= \operatorname*{argmax}_{\boldsymbol{r}} \min_d \left[V^{\pi(\boldsymbol{r})}(\phi_d) - \hat{V}^E(\phi_d)\right].\end{aligned}$$

Hence, the optimal policy induced by $\boldsymbol{r}_{\text{MAP}}$ is equivalent to π_{MWAL} since $\boldsymbol{r}_{\text{MAP}} = \boldsymbol{r}_{\text{MWAL}}$. ■

Lemma 4: The policy sought by the policy matching algorithm [11] is equivalent to an optimal policy on the reward function which is the MAP estimate with the uniform prior and the likelihood using $f_J(\Xi; \boldsymbol{r}) = -\sum_{s,a}\hat{x}_E(s, a)(J(s, a; \boldsymbol{r}) - \hat{\pi}_E(s, a))^2$, where $\hat{x}_E(s, a)$ is the empirical estimate of state-action visitation counts of expert's policy π_E and $J(s, a; \boldsymbol{r})$ is a smooth mapping from reward function $\boldsymbol{r}$ to a greedy policy, such as the soft-max function.

Proof: The policy matching algorithm seeks the policy $\pi_{\text{PM}} = J(\boldsymbol{r}_{\text{PM}})$ such that

$$\boldsymbol{r}_{\text{PM}} = \operatorname*{argmin}_{\boldsymbol{r}} \sum_{s,a}\hat{x}_E(s, a)(J(s, a; \boldsymbol{r}) - \hat{\pi}_E(s, a))^2.$$

The MAP estimate of the reward function with the uniform prior and the likelihood using f_J is computed as

$$\begin{aligned}\boldsymbol{r}_{\text{MAP}} &= \operatorname*{argmax}_{\boldsymbol{r}} P(\boldsymbol{r}|\Xi) \\ &= \operatorname*{argmax}_{\boldsymbol{r}} f_J(\Xi; \boldsymbol{r}) \\ &= \operatorname*{argmin}_{\boldsymbol{r}} \sum_{s,a}\hat{x}_E(s, a)(J(s, a; \boldsymbol{r}) - \hat{\pi}_E(s, a))^2.\end{aligned}$$

Hence, $\boldsymbol{r}_{\text{MAP}} = \boldsymbol{r}_{\text{PM}}$ and the optimal policy induced by $\boldsymbol{r}_{\text{MAP}}$ is equivalent to π_{PM}. ■

Lemma 5: The reward function sought by the MaxEnt algorithm [12] is equivalent to the MAP estimate with the uniform prior and the likelihood using $f_E(\Xi; \boldsymbol{r}) = \log \mathcal{P}_{\text{MaxEnt}}(\Xi|\boldsymbol{r})$ where $\mathcal{P}_{\text{MaxEnt}}$ is the distribution for the behavior data (trajectory or path) satisfying the principle of maximum entropy.

Proof: The MaxEnt algorithm seeks the reward function defined by

$$\begin{aligned}\boldsymbol{r}_{\text{MaxEnt}} &= \operatorname*{argmax}_{\boldsymbol{r}} \log \mathcal{P}_{\text{MaxEnt}}(\Xi|\boldsymbol{r}) \\ &= \operatorname*{argmax}_{\boldsymbol{r}} \sum_{\xi \in \Xi} \log \mathcal{P}_{\text{MaxEnt}}(\xi|\boldsymbol{r})\end{aligned}$$

where $\mathcal{P}_{\text{MaxEnt}}(\xi|\boldsymbol{r}) = \frac{1}{Z(\boldsymbol{r})}\exp(\boldsymbol{r}^{\intercal}\boldsymbol{\mu}(\xi))$, $\boldsymbol{\mu}(\xi)$ is the empirical estimate of the feature expectation from trajectory ξ, and

$Z(\boldsymbol{r})$ is a normalization constant. The MAP estimate with the uniform prior and the likelihood using f_E is computed as

$$\begin{aligned}\boldsymbol{r}_{\text{MAP}} &= \operatorname*{argmax}_{\boldsymbol{r}} P(\boldsymbol{r}|\Xi) \\ &= \operatorname*{argmax}_{\boldsymbol{r}} f_E(\Xi;\boldsymbol{r}) \\ &= \operatorname*{argmax}_{\boldsymbol{r}} \log \mathcal{P}_{\text{MaxEnt}}(\Xi|\boldsymbol{r}).\end{aligned}$$

The MAP estimate is thus equivalent to $\boldsymbol{r}_{\text{MaxEnt}}$. ■

C. Proof of Theorem 2

Proof: Let $C(\pi)$ be the reward optimality region with respect to π

$$\boldsymbol{v}^*(\boldsymbol{r}) = \boldsymbol{v}^{\pi}(\boldsymbol{r}) = (\boldsymbol{I} - \gamma \boldsymbol{T}^{\pi})^{-1}\boldsymbol{E}^{\pi}\boldsymbol{r}$$

for any $\boldsymbol{r} \in C(\pi)$, so $\boldsymbol{v}^*(\boldsymbol{r})$ is linear with respect to $\boldsymbol{r}$. For each and every $\boldsymbol{r}_1$, $\boldsymbol{r}_2$, and $0 \leq \mu \leq 1$

$$\begin{aligned}\boldsymbol{v}^*(\mu\boldsymbol{r}_1 + (1-\mu)\boldsymbol{r}_2) &= \boldsymbol{H}^{\pi}(\mu\boldsymbol{r}_1 + (1-\mu)\boldsymbol{r}_2) \\ &= \mu\boldsymbol{H}^{\pi}\boldsymbol{r}_1 + (1-\mu)\boldsymbol{H}^{\pi}\boldsymbol{r}_2 \\ &= \mu\boldsymbol{v}^{\pi}(\boldsymbol{r}_1) + (1-\mu)\boldsymbol{v}^{\pi}(\boldsymbol{r}_2) \\ &\leq \mu\boldsymbol{v}^*(\boldsymbol{r}_1) + (1-\mu)\boldsymbol{v}^*(\boldsymbol{r}_2)\end{aligned}$$

where $\boldsymbol{H}^{\pi} = (\boldsymbol{I} - \gamma\boldsymbol{T}^{\pi})^{-1}\boldsymbol{E}^{\pi}$ and π is an optimal policy for $\mu\boldsymbol{r}_1 + (1-\mu)\boldsymbol{r}_2$. Thus, $\boldsymbol{v}^*(\boldsymbol{r})$ is convex. In the same manner, we can also show that $\boldsymbol{q}^*(\boldsymbol{r})$ is convex using the definition $\boldsymbol{q}^{\pi}(\boldsymbol{r}) = \boldsymbol{r} + \gamma\boldsymbol{T}\boldsymbol{E}^{\pi}\boldsymbol{q}^{\pi}(\boldsymbol{r})$. ■

D. Proof of Theorem 3

Proof: Let $C(\pi)$ be the reward optimality region with respect to π. Since $\boldsymbol{v}^*(\boldsymbol{r}) = \boldsymbol{v}^{\pi}(\boldsymbol{r}) = (\boldsymbol{I} - \gamma\boldsymbol{T}^{\pi})^{-1}\boldsymbol{E}^{\pi}\boldsymbol{r}$ is linear for any $\boldsymbol{r} \in C(\pi)$, $\boldsymbol{v}^*(\boldsymbol{r})$ is differentiable and $\nabla_{\boldsymbol{r}}\boldsymbol{v}^*(\boldsymbol{r}) = (\boldsymbol{I} - \gamma\boldsymbol{T}^{\pi})^{-1}\boldsymbol{E}^{\pi}$ when $\boldsymbol{r}$ is strictly inside the region. On the boundary, $\nabla_{\boldsymbol{r}}\boldsymbol{v}^{\pi}(\boldsymbol{r})$ is a sub-gradient of $\boldsymbol{v}^*(\boldsymbol{r})$ since the function is convex from Theorem 2 and thus

$$\nabla_{\boldsymbol{r}}\boldsymbol{v}^{\pi}(\boldsymbol{r})(\boldsymbol{r} - \boldsymbol{r}') \leq \boldsymbol{v}^*(\boldsymbol{r}) - \boldsymbol{v}^*(\boldsymbol{r}')$$

for any $\boldsymbol{r}'$. In the same manner, we can also show that $\boldsymbol{q}^*(\boldsymbol{r})$ is differentiable with $\nabla_{\boldsymbol{r}}\boldsymbol{q}^*(\boldsymbol{r}) = (\boldsymbol{I} - \gamma\boldsymbol{T}\boldsymbol{E}^{\pi})^{-1}$ strictly inside reward optimality regions and $\nabla_{\boldsymbol{r}}\boldsymbol{q}^{\pi}(\boldsymbol{r})$ is a sub-gradient on the boundaries. ■

REFERENCES

[1] S. Russell, "Learning agents for uncertain environments (extended abstract)," in *Proc. 11th Annu. Conf. Comput. Learn. Theory*, 1998, pp. 101–103.

[2] H. Hattori, Y. Nakajima, and T. Ishida, "Learning from humans: Agent modeling with individual human behaviors," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 41, no. 1, pp. 1–9, Jan. 2011.

[3] P. Abbeel, A. Coates, and A. Y. Ng, "Autonomous helicopter aerobatics through apprenticeship learning," *Int. J. Robot. Res.*, vol. 29, no. 13, pp. 1608–1639, 2010.

[4] B. D. Ziebart, A. L. Maas, A. K. Dey, and J. A. Bagnell, "Navigate like a cabbie: Probabilistic reasoning from observed context-aware behavior," in *Proc. 10th Int. Conf. Ubiquit. Comput.*, Seoul, Korea, 2008, pp. 322–331.

[5] Z. Erkin, M. D. Bailey, L. M. Maillart, A. J. Schaefer, and M. S. Roberts, "Eliciting patients' revealed preferences: An inverse Markov decision process approach," *Decis. Anal.*, vol. 7, no. 4, pp. 358–365, 2010.

[6] S. Chandramohan, M. Geist, F. Lefevre, and O. Pietquin, "User simulation in dialogue systems using inverse reinforcement learning," in *Proc. Interspeech*, 2011.

[7] S. J. Lee and Z. Popović, "Learning behavior styles with inverse reinforcement learning," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 1–7, 2010.

[8] A. Vogel, D. Ramachandran, R. Gupta, and A. Raux, "Improving hybrid vehicle fuel efficiency using inverse reinforcement learning," in *Proc. 26th AAAI Conf. Artif. Intell.*, 2012.

[9] A. Y. Ng and S. Russell, "Algorithms for inverse reinforcement learning," in *Proc. 17th Int. Conf. Mach. Learn.*, 2000, pp. 663–670.

[10] N. D. Ratliff, J. A. Bagnell, and M. A. Zinkevich, "Maximum margin planning," in *Proc. 23rd Int. Conf. Mach. Learn.*, Pittsburgh, PA, USA, 2006, pp. 729–736.

[11] G. Neu and C. Szepesvári, "Apprenticeship learning using inverse reinforcement learning and gradient methods," in *Proc. 23rd Conf. Uncertainty Artif. Intell.*, 2007, pp. 295–302.

[12] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *Proc. 23rd AAAI Conf. Artif. Intell.*, 2008, pp. 1433–1438.

[13] D. Ramachandran and E. Amir, "Bayesian inverse reinforcement learning," in *Proc. 20th Int. Joint Conf. Artif. Intell.*, 2007, pp. 2586–2591.

[14] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, pp. 1–8.

[15] H. A. Simon, "A behavioral model of rational choice," *Quart. J. Econ.*, vol. 69, no. 1, pp. 99–118, 1955.

[16] J. G. March, "Bounded rationality, ambiguity, and the engineering of choice," *Bell J. Econ.*, vol. 9, no. 2, pp. 587–608, 1978.

[17] G. Gigerenzer, *Bounded Rationality: The Adaptive Toolbox*. Cambridge, MA, USA: MIT Press, 2002.

[18] U. Syed and R. E. Schapire, "A game-theoretic approach to apprenticeship learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 20. Vancouver, BC, Canada, 2007, pp. 1449–1456.

[19] C. A. Rothkopf and C. Dimitrakakis, "Preference elicitation and inverse reinforcement learning," in *Proc. 22nd Eur. Conf. Mach. Learn.*, 2011, pp. 34–48.

[20] J. Choi and K.-E. Kim, "MAP inference for Bayesian inverse reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 24. Granada, Spain, 2011, pp. 1989–1997.

[21] T. Lotan, "Modeling discrete choice behavior based on explicit information integration and its application to the route choice problem," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 28, no. 1, pp. 100–114, Jan. 1998.

[22] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY, USA: Wiley, 1994.

[23] P. Dayan and G. E. Hinton, "Using expectation-maximization for reinforcement learning," *Neural Comput.*, vol. 9, no. 2, pp. 271–278, 1997.

[24] I. Brocas and J. D. Carrillo, *The Psychology of Economic Decisions*. Oxford, U.K.: Oxford Univ. Press, 2003.

[25] D. Kahneman, "Maps of bounded rationality: Psychology for behavioral economics," *Amer. Econ. Rev.*, vol. 93, no. 5, pp. 1449–1475, 2003.

[26] G. Casella and R. Berger, *Statistical Inference*. Pacific Grove, CA, USA: Duxbury Press, 2001.

[27] U. Syed, M. Bowling, and R. E. Schapire, "Apprenticeship learning using linear programming," in *Proc. 25th Int. Conf. Mach. Learn.*, Helsinki, Finland, 2008, pp. 1032–1039.

[28] A. Boularias and B. Chaib-draa, "Bootstrapping apprenticeship learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 23. Vancouver, BC, Canada, 2010.

[29] H. D. Dixon, *Surfing Economics: Essays for the Inquiring Economist*. Basingstokke, U.K.: Palgrave MacMillan, 2001.

[30] M. Piorkowski, N. Sarafijanovoc-Djukic, and M. Grossglauser, "A parsimonious model of mobile partitioned networks with clustering," in *Proc. 1st Int. Conf. Commun. Syst. Netw. Workshops*, 2009, pp. 1–10.

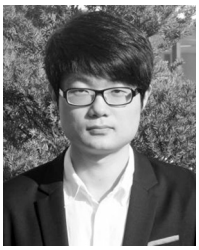
[31] J. Letchner, J. Krumm, and E. Horvitz, "Trip router with individualized preferences (TRIP): Incorporating personalization into route planning," in *Proc. 21st Nat. Conf. Artif. Intell.*, 2006, pp. 1795–1800.

[32] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, CA, USA: Athena Scientific, 1995.

[33] T. G. Dietterich, "Hierarchical reinforcement learning with the MAXQ value function decomposition," *J. Artif. Intell. Res.*, vol. 13, no. 1, pp. 227–303, 2000.

[34] M. Ghavamzadeh and S. Mahadevan, "Hierarchical average reward reinforcement learning," *J. Mach. Learn. Res.*, vol. 8, pp. 2629–2669, Nov. 2007.

[35] X. Xu, C. Liu, S. X. Yang, and D. Hu, "Hierarchical approximate policy iteration with binary-tree state space decomposition," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 1863–1877, Dec. 2011.



Jaedeug Choi (S'12) received the B.S. degree in computer science and engineering from the Pohang University of Science and Technology (POSTECH), Pohang, Korea, in 2008, and the M.S. and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 2009 and 2013, respectively.

While actively advancing his career as a Post-Doctoral Scholar back at POSTECH, his life was sadly claimed by an accident on October 30, 2013. His research interests included models and algorithms for machine learning, reinforcement learning, partially observable Markov decision processes, inverse reinforcement learning, and Bayesian nonparametrics.

Kee-Eung Kim (M'10) received the B.S. degree in computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1995, and the M.Sc. and Ph.D. degrees in computer science from Brown University, Providence, RI, USA, in 1998 and 2001, respectively.

From 2001 to 2006, he was a Senior Software Engineer at Samsung SDS, Korea, and a Senior Research Staff Member at Samsung Advanced Institute of Technology, Korea. In 2006, he joined the faculty of Computer Science Department at KAIST, where he is currently an Associate Professor. His research interests included representations and algorithms for sequential decision making problems in artificial intelligence and machine learning, Markov decision processes, and reinforcement learning.