The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Efficient Probabilistic Performance Bounds for Inverse Reinforcement Learning

Daniel S. Brown, Scott Niekum
Department of Computer Science
University of Texas at Austin
{dsbrown,sniekum}@cs.utexas.edu

Abstract

In the field of reinforcement learning there has been recent progress towards safety and high-confidence bounds on policy performance. However, to our knowledge, no practical methods exist for determining high-confidence policy performance bounds in the inverse reinforcement learning setting—where the true reward function is unknown and only samples of expert behavior are given. We propose a sampling method based on Bayesian inverse reinforcement learning that uses demonstrations to determine practical high-confidence upper bounds on the α-worst-case difference in expected return between any evaluation policy and the optimal policy under the expert's unknown reward function. We evaluate our proposed bound on both a standard grid navigation task and a simulated driving task and achieve tighter and more accurate bounds than a feature count-based baseline. We also give examples of how our proposed bound can be utilized to perform risk-aware policy selection and risk-aware policy improvement. Because our proposed bound requires several orders of magnitude fewer demonstrations than existing high-confidence bounds, it is the first practical method that allows agents that learn from demonstration to express confidence in the quality of their learned policy.

Introduction

There is a growing interest in safety and risk-sensitive metrics for machine learning and artificial intelligence systems, especially for systems that interact with their environment (Garcıa and Fernández 2015; Amodei et al. 2016; Thomas et al. 2017). Risk-aware approaches have been recently proposed and applied to many different problems including planning in Markov decision processes (Chow et al. 2015), physical search problems (Brown et al. 2016), reinforcement learning (Tamar, Glassner, and Mannor 2015; Garcıa and Fernández 2015), and imitation learning (Santara et al. 2017); however, to the best of our knowledge, no one has investigated how to obtain sample-efficient, risk-aware confidence bounds on the performance of a policy under an unknown reward function, as is the case when learning from demonstrations.

Learning from demonstration (LfD) is a popular method to learn a skill or policy by simply observing demonstrations from an expert (Argall et al. 2009). One popular variant of LfD is Inverse Reinforcement Learning (IRL) (Ng and Russell 2000) where the goal is to infer a reward function that explains the demonstrated behavior. LfD techniques based on IRL have potential applications in many settings such as manufacturing, home and hospital care, and autonomous driving. In these types of real-world settings it is important, and perhaps critical, to provide performance bounds on an agent's learned policy. For example, consider a hospital assistant robot that has learned from demonstrations how to lift a patient out of bed. Before deploying this learned policy, we would want to provide a high-confidence bound on the difference in performance between the robot's learned policy and the optimal policy under the expert's reward. If this bound on policy loss is too high, then the robot could request additional demonstrations until, with high confidence, its policy loss with respect to the optimal policy is within some allowable error margin.

We propose a general method for obtaining high-confidence performance bounds in the inverse reinforcement learning setting—where the true reward function is unknown and only samples of expert behavior are given. Given demonstrated trajectories of a task, our goal is to allow an agent to bound the difference in expected return between the agent's own policy and the optimal policy for the task, under the expert's unknown reward function. Because the problem of Inverse Reinforcement Learning is ill-posed (there are an infinite number of reward functions that result in the same optimal behavior), we seek a risk-sensitive bound on this performance difference that takes into account the uncertainty in the posterior distribution over reward functions, conditioned on the demonstrations.

We perform Markov chain Monte Carlo sampling using Bayesian Inverse Reinforcement Learning (Ramachandran and Amir 2007) to sample likely reward functions given the demonstrations. Using these sampled reward functions, we compute samples of the expected return difference between the optimal policy under the expert's reward function and the agent's policy. These samples are then used to calculate a $(1-\delta)$ probabilistic upper bound on the α-worst-case policy loss. We obtain this bound without knowing the expert's policy or true reward function.

Our main contributions are (1) we formalize the problem of high-confidence policy evaluation in the inverse reinforcement learning domain; (2) we present the first practical method for obtaining high-confidence bounds on the α-worst-case difference in expected return between any evaluation policy and the optimal policy under the expert's unknown reward function; (3) we evaluate our proposed bound on standard grid navigation and simulated driving tasks and demonstrate a significant improvement over existing statistical bounds and an empirical baseline based on feature counts; and (4) we give examples of how our proposed bound enables risk-aware policy ranking and risk-aware policy improvement, given only demonstrations of a task.

Preliminaries

Markov decision processes

A Markov decision process (MDP) is defined as a tuple $\langle S, A, T, R, \gamma, S_0 \rangle$ where S is the set of states, A is the set of actions, $T : S \times A \times S \to [0, 1]$ is the transition function, $R : S \to \mathbb{R}$ is the reward function, $\gamma \in [0, 1)$ is the discount factor, and S_0 is the initial state distribution.

A policy π is a mapping from states to a probability distribution over actions. The value of a policy π under reward function R is the expected return of that policy and is denoted as $V_R^\pi = \mathbb{E}_{s_0 \sim S_0}[\sum_{t=0}^{\infty} \gamma^t R(s_t)|\pi]$. The value of executing policy π starting at state $s \in S$ is defined as $V_R^\pi(s) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t)|\pi, s_0 = s]$. Given a reward function R, the Q-value of a state-action pair (s, a) is defined as $Q_R^\pi(s, a) = R(s) + \gamma \sum_{s' \in S} T(s, a, s') V_R^\pi(s')$. We denote $V_R^* = \max_\pi V_R^\pi$ and $Q_R^*(s, a) = \max_\pi Q_R^\pi(s, a)$.

As is common in the literature (Abbeel and Ng 2004; Ziebart et al. 2008), we assume that the reward function can be expressed as a linear combination of features, so that $R(s) = w^T \phi(s)$ where $w \in \mathbb{R}^k$ is the k-dimensional vector of feature weights. Thus, we can write the value of a policy as $V_R^\pi = \mathbb{E}_{s_0 \sim S_0}[\sum_{t=0}^{\infty} \gamma^t w^T \phi(s_t)|\pi] = w^T \mu(\pi)$, where $\mu(\pi) = \mathbb{E}_{s_0 \sim S_0}[\sum_{t=0}^{\infty} \gamma^t \phi(s_t)|\pi]$ are the expected feature counts. Note that this does not affect the expressiveness of the reward function since ϕ can be a non-linear function. Given ϕ, the reward function is fully specified by the feature weights w. Thus, we refer to the feature weights w and the reward function R interchangeably.

Bayesian inverse reinforcement learning

In IRL we are given an MDP without a reward function, denoted MDP\R. Given a set of demonstrations, $D = \{(s_1, a_1), \ldots, (s_m, a_m)\}$, consisting of state-action pairs, the IRL problem is to recover the reward function, R^*, of the demonstrator. Because this problem is ill-posed, IRL algorithms use a variety of heuristics and simplifying assumptions to find an estimate of R^* (Gao et al. 2012).

Bayesian IRL (BIRL) (Ramachandran and Amir 2007) seeks to estimate the posterior over reward functions given demonstrations, $P(R|D) \propto P(D|R)P(R)$. BIRL makes the assumption that the demonstrator is following a softmax policy, resulting in the likelihood function

$$P(D|R) = \prod_{(s,a) \in D} P((s, a)|R) = \prod_{(s,a) \in D} \frac{e^{cQ_R^*(s,a)}}{\sum_{b \in A} e^{cQ_R^*(s,b)}} \tag{1}$$

where $Q_R^*(s, a)$ is the optimal Q-value function for reward R, and c is a parameter representing the confidence in the demonstrator's optimality. Equation 1 gives greater likelihood to rewards for which the actions taken by the expert have higher Q-values than the alternative actions.

The softmax distribution over actions is commonly used as a likelihood function in IRL (Babes et al. 2011; Levine, Popovic, and Koltun 2011; Michini and How 2012a; Rothkopf and Ballard 2013) and has been empirically shown to be an effective model of human behavior, enabling accurate learning from human demonstrations (Lopes, Melo, and Montesano 2007; Kim and Pineau 2016) and prediction of human actions (Baker, Saxe, and Tenenbaum 2009; Karasev et al. 2016).

The BIRL algorithm uses Markov chain Monte Carlo (MCMC) sampling to sample from the posterior $P(R|D)$. Feature weights are sampled according to a proposal distribution, and for each sample the MDP is solved to obtain the sample's likelihood and determine the transition probabilities within the Markov chain. For each new sample, the resulting MDP can typically be quickly solved by starting with the policy from the previous MDP and using only a few steps of policy iteration (Ramachandran and Amir 2007). An estimate of the expert's reward function can be found by averaging the feature weights in the chain to obtain the mean reward function (Ramachandran and Amir 2007) or by using the maximum a posteriori (MAP) estimate (Choi and Kim 2011). Some of the advantages of BIRL, compared to many other IRL algorithms, are (1) it finds a distribution over likely reward functions, (2) D can contain partial demonstrations or even non-contiguous state action pairs, and (3) it works with sub-optimal demonstrations.

The choice of the prior allows domain knowledge to be inserted into the IRL algorithm. Ramachandran et al. (2007) give several possibilities such as a uniform, Gaussian, or Beta prior. For the remainder of this paper we assume the prior is uniform. Evaluating the effects of alternative priors is left to future work.

Problem Definition

We assume that we are given an MDP\R and samples $D = \{(s_1, a_1), \ldots, (s_m, a_m)|(s_i, a_i) \sim \pi_{\text{demo}}\}$ of state-action pairs from a demonstrator's policy π_{demo}. We make the common assumption (Abbeel and Ng 2004; Ramachandran and Amir 2007) that the demonstrator attempts to maximize total return under the reward R^* by executing a possibly suboptimal, stationary policy π_{demo}. Given any evaluation policy π_{eval}, we are interested in the following general problem:

High-confidence policy evaluation for LfD: Given an MDP\R, an evaluation policy π_{eval}, and a set of demonstrations, D, find a high-confidence upper bound on the policy loss incurred by using π_{eval} in place of π^*, where π^* is the optimal policy for the demonstrator's reward function, R^*.

We define policy loss using the *Expected Value Difference* (EVD) of π_{eval} under the true reward R^*, defined as

$$\text{EVD}(\pi_{\text{eval}}, R^*) = V_{R^*}^* - V_{R^*}^{\pi_{\text{eval}}}. \tag{2}$$

We use EVD because it is a natural way to measure the performance difference between two policies and it is a common metric for evaluating IRL algorithms (Ramachandran and Amir 2007; Levine, Popovic, and Koltun 2011; Choi and Kim 2011; Wulfmeier, Ondruska, and Posner 2015). Note that the evaluation policy can be any policy, including a hand-tuned policy or a policy learned through reinforcement learning on a different task with a known reward function; however, the most natural form of the evaluation policy is a policy learned from the demonstrations, D.

We seek to bound the difference in expected return between the evaluation policy π_{eval} and π^*, the policy that is optimal with respect to the demonstrator's reward R^*. However, because an optimal policy is invariant to any non-negative scaling of the reward function, bounding EVD is ill-posed, as we can multiply the feature weights w by any $c > 0$ to scale EVD to be anywhere in the range $[0, \infty)$. To avoid this scaling issue we make the common assumption that $\|w\|_1 = 1$ (Syed and Schapire 2008; Pirotta and Restelli 2016). Note, that this assumption only eliminates the trivial all-zero reward function as a potential solution—all other reward functions can be appropriately normalized. While setting $\|w\|_1 = 1$ eliminates the invariance to scaling factors and bounds the magnitude of the EVD, there can still be infinitely many rewards that induce any optimal policy, resulting in infinitely many possible values of $\text{EVD}(\pi_{\text{eval}}, R^*)$. Thus, to obtain an upper bound on $\text{EVD}(\pi_{\text{eval}}, R^*)$ we need to address this uncertainty.

As we show in the following section, one way to address this uncertainty over the demonstrator's true reward is to compute an absolute worst-case policy loss bound using feature counts. However, as we show in the evaluation section, this type of worst-case bound is sensitive to adversarial reward functions that are highly unlikely given the demonstrations, often resulting in loose bounds. Thus, rather than focusing on absolute worst-case, we focus on computing a probabilistic upper bound on the α-worst-case value of $\text{EVD}(\pi_{\text{eval}}, R)$, where $R \sim P(R|D)$.

The α-worst-case value of a random variable is often referred to in finance as the α-Value at Risk (Jorion 1997). We use the notation of Tamar et al. (Tamar, Glassner, and Mannor 2015) and formally define the *α-Value-at-Risk* of a random variable Z as

$$\nu_\alpha(Z) = F_Z^{-1}(\alpha) = \inf\{z : F_Z(z) \geq \alpha\} \tag{3}$$

where $\alpha \in (0, 1)$ is the quantile level and $F_Z(z) = Pr(Z \leq z)$ is the cumulative distribution function of Z.

The specific problem that we address is the following:

Risk-aware policy evaluation for LfD: Given an MDP\R, any evaluation policy π_{eval}, and a set of demonstrations, D, find a $(1 - \delta)$ confidence upper bound on $\nu_\alpha(\text{EVD}(\pi_{\text{eval}}, R))$, where $R \sim P(R|D)$.

Note that α defines the sensitivity to risk, while $(1 - \delta)$ represents our confidence in our estimate of the α-VaR. Thus, while $(1 - \delta)$ is typically always high (e.g., 0.95), α can take on a range of values depending on the possibility of catastrophic failure in the domain and the risk-aversion of the end-user. In practice, $\alpha \geq 0.9$ is commonly used for VaR applications (Jorion 1997).

Worst-Case Bound

Before we give the full details of our approach, we first derive a simple worst-case bound based on feature counts that we use as a baseline. As a reminder, we use the notation $\mu(\pi) = \mathbb{E}_{s_0 \sim S_0}[\sum_{t=0}^{\infty} \gamma^t \phi(s_t)|\pi]$ to represent the expected feature counts of policy π.

Given any evaluation policy π_{eval}, Abbeel and Ng (2004) showed that if we assume $\phi(s) : S \to [0, 1]^k$, $\|w\|_1 \leq 1$, and know the demonstrator's expected feature counts $\mu^* = \mu(\pi_{\text{demo}})$, then $\|\mu^* - \mu(\pi_{\text{eval}})\|_2 \leq \epsilon$ implies that

$$V_R^{\pi_{\text{demo}}} - V_R^{\pi_{\text{eval}}} = w^T(\mu^* - \mu(\pi_{\text{eval}})) \leq \epsilon$$

for any reward function $R(s) = w^T \phi(s)$. If π_{demo} is optimal with respect to the demonstrator's reward function, R^*, then

$$w^T(\mu^* - \mu(\pi_{\text{eval}})) = \text{EVD}(\pi_{\text{eval}}, R^*) \leq \epsilon$$

and $\|\mu^* - \mu(\pi_{\text{eval}})\|_2$ gives an upper bound on $\text{EVD}(\pi_{\text{eval}}, R^*)$.

We now derive an even tighter bound. First, note that the worst-case policy loss is the objective value of the following maximization problem

$$\max_w \quad w^T(\mu^* - \mu(\pi_{\text{eval}})) \tag{4}$$

$$\text{subject to} \quad \|w\|_1 = 1. \tag{5}$$

The solution is to put all of our budget for w on the feature with maximal feature count difference, giving the solution $\|\mu^* - \mu(\pi_{\text{eval}})\|_\infty$. Because the two-norm is always lower bounded by the infinity-norm, this bound will be tighter than the bound proposed by Abbeel and Ng (2004).

Note that in practice we do not know μ^*, but we can use demonstrated trajectories to estimate of the demonstrator's expected feature counts as

$$\hat{\mu}^* = \frac{1}{|D|} \sum_{i=1}^{|D|} \sum_{t=0}^{\infty} \gamma^t \phi(s_t^{(i)}), \tag{6}$$

where i indexes over the trajectories and t over the state sequence contained in each demonstrated trajectory. We define the empirical *worst-case feature count bound* as

$$\text{WFCB}(\pi_{\text{eval}}, D) = \|\hat{\mu}^* - \mu(\pi_{\text{eval}})\|_\infty. \tag{7}$$

Note that for this bound to be a guaranteed upper bound on $\text{EVD}(\pi_{\text{eval}}, R^*)$, π_{demo} must be optimal and the empirical estimate of the expert's feature counts, $\hat{\mu}^*$, may require a large number of demonstrations to converge to μ^* (Abbeel and Ng 2004; Syed and Schapire 2008). Other limitations of this bound are that it does not work with partial demonstrations and that it is based on an adversarial reward function that may be extremely unlikely given the demonstrations.

EVD Value-at-Risk Bound

The worst-case feature count bound described in the previous section only requires sampled trajectories from the expert, but completely ignores the structure of the problem and the actions taken by the demonstrator—giving a worst-case bound that will likely be overly pessimistic. Our goal is to obtain a high-confidence probabilistic worst-case bound that focuses on likely reward functions given the demonstrations.

We seek a probabilistic confidence bound on the α-Value at Risk of the EVD(π_{eval}, R^*) for any given evaluation policy π_{eval}. We note that using the EVD rather than a standard feature count bound, as discussed in the previous section, is desirable for two main reasons. The first reason is that it works well with partial, noisy demonstrations. This is because EVD compares the evaluation policy against the optimal policy for reward R, not the actual states visited by the potentially sub-optimal demonstrator. Second, the EVD explicitly takes into account the initial state distribution. Thus, EVD measures the generalizability error of an evaluation policy by evaluating the expected return over all states with support under S_0, even if demonstrations have only been sampled from a small number of possible initial states.

To bound the α-quantile worst-case EVD(π_{eval}, R^*) we use samples from the posterior $P(R|D)$. Thus, we seek to calculate $\nu_\alpha(Z)$ where $Z = EVD(\pi_{\text{eval}}, R)$ for $R \sim P(R|D)$. As motivated in the Problem Definition, we assume $\|w\|_1 = 1$. Thus, to find $P(R|D)$ we use a modified version of the BIRL Policy Walk Algorithm (Ramachandran and Amir 2007) that ensures that our proposal samples of w during MCMC stay on the L1-norm unit ball. Details are given in the full paper (Brown and Niekum 2017). Using MCMC, we generate a sequence of sampled rewards $\mathcal{R} = \{R : R \sim P(R|D)\}$ from the posterior distribution over reward functions given the demonstrations. For each sample $R_i \in R$ we then calculate

$$Z_i = \text{EVD}(\pi_{\text{eval}}, R_i) = V^*_{R_i} - V^{\pi_{\text{eval}}}_{R_i} \tag{8}$$

giving us samples from the posterior distribution over expected value differences.

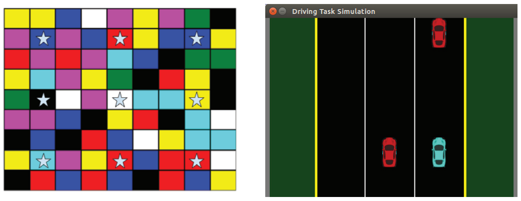
To obtain a point estimate of α-VaR we can sort the resulting samples of Z in ascending order to obtain the order statistics Y, and then take the α-quantile. However, this does not take into account the number of samples or our confidence in this point estimate. Instead of using a point estimate, we compute a single-sided $(1-\delta)$ confidence bound on the α-VaR. Given a sample Z_i, we have that $P(Z_i < \nu_\alpha(Z)) = \alpha$. Thus, for any order statistic Y_j, we can use the normal approximation of the binomial distribution to obtain

$$\begin{aligned} P(\nu_\alpha(Z) \leq Y_j) &= \sum_{i=1}^{j} \binom{N}{i} \alpha^i (1-\alpha)^{N-i} \qquad (9) \\ &\approx F_{\mathcal{N}}\left(j + \frac{1}{2} \mid N\alpha, N\alpha(1-\alpha)\right). \end{aligned}$$

where $F_{\mathcal{Z}}$ is the CDF of the normal distribution with $\mu = N\alpha$ and $\sigma^2 = N\alpha(1-\alpha)$ and 1/2 is added to the index j as a continuity correction (Hollander and Wolfe 1999). To obtain the index k of the order statistic such that $P(\nu_\alpha(Z) \leq Y_k) \geq (1-\delta)$ we invert Equation 9 using the inverse of the standard normal CDF, $F_{\mathcal{N}}^{-1}$, to get $k = \lceil N\alpha + F_{\mathcal{N}}^{-1}(1-\delta)\sqrt{N\alpha(1-\alpha)} - \frac{1}{2} \rceil$. Our full approach is summarized in

Algorithm 1 $(1-\delta)$ Confidence Bound on the α-Value-at-Risk

```
input: MDP\R, $\pi_{\text{eval}}$, $D$, $c$, $\alpha$, $\delta$
$\mathcal{R} \leftarrow$ BIRL(MDP\R, $D$, $c$)    ▷ sample from posterior using L1-unit norm walk
for $R_i \in \mathcal{R}$ do
    $Z_i = V^*_{R_i} - V^{\pi_{\text{eval}}}_{R_i}$    ▷ compute policy loss
$Y$ = sort($Z$)    ▷ sort into ascending order statistics
$k = \lceil N\alpha + F_{\mathcal{N}}^{-1}(1-\delta)\sqrt{N\alpha(1-\alpha)} - \frac{1}{2} \rceil$    ▷ index of $(1-\delta)$ confidence bound on $\alpha$-VaR
return $Y_k$
```



(a) Grid world navigation (b) Driving simulation

Figure 1: (a) Example of random grid world navigation task with colors representing random features and initial states denoted by stars. (b) Snapshot of driving simulation. Agent must learn to safely drive blue car through traffic.

Algorithm 1. The algorithm has three hyperparameters: c defines the confidence in the optimality of the demonstrations, α defines the risk-sensitivity, and $(1-\delta)$ represents the desired confidence level on the estimate of the α-VaR.

The advantages of our approach are as follows: (1) our proposed bound takes full advantage of all of the information contained in the transition dynamics and demonstrations to focus on reward functions that are likely given the demonstrations, (2) it does not require optimal demonstrations, (3) it inherits from BIRL the ability to work with partial demonstrations, even disjoint state-action pairs, and (4) it allows for domain knowledge in the form of a prior.

Empirical results

For our proposed confidence bound to be useful, it needs to meet several criteria: (1) the upper bound should be accurate with high-confidence (rarely underestimating the true expected value difference), (2) the bound should be tighter than the worst-case bound derived above, and (3) the previous two criteria should be true even when given a small number of demonstrations. We use both a standard grid world navigation task (Abbeel and Ng 2004; Ramachandran and Amir 2007; Choi and Kim 2011) and a simulated driving task (Abbeel and Ng 2004; Syed and Schapire 2008; Cohn, Durfee, and Singh 2011) to validate that our proposed bound satisfies these criteria. Examples of these tasks are shown in Figure 1. We compare our high-confidence α-VaR bound with the worst-case feature count bound (WFCB) defined in Equation 7. All results for α-VaR bounds are reported as 95% confidence bounds ($\delta = 0.05$).

Grid world navigation task

We first empirically evaluate our approach on a suite of 9x9 grid world navigation tasks where the cost of traveling on different terrains is unknown and must be inferred from demonstrations. The available actions are up, down, left and right. Transitions are noisy with an 70% chance of moving in the desired direction and 30% chance of going in one of the directions perpendicular to the chosen direction. There are 8 binary features with one feature active per grid cell. To show that our results are not an artifact of a specific reward function, we evaluate our method over many random grid worlds, each with a randomly chosen ground truth reward. We use $\gamma = 0.9$ and an initial state distribution S_0 that is uniform over 9 different states spread across the grid as shown in Figure 1(a). When generating M demonstrations we select the initial states in a round-robin fashion from the support of S_0. However, when measuring accuracy and bound errors, we compare with the true expected value difference over the full initial state distribution.

Infinite horizon grid navigation Our first task is an infinite horizon grid world navigation task with no terminal states. To evaluate different bounding methods we generated 200 random 9x9 worlds with random features each grid cell. For each world we generated a random feature weight vector w from the L1-unit norm ball. To generate demonstrations we solve the MDP using the random ground truth reward to find the optimal policy and use this policy to generate trajectories of length 100. We set the evaluation policy to be the optimal policy under the MAP reward function found using BIRL. Because the demonstrations in this experiment are perfect, we set the BIRL confidence parameter to a large value ($c = 100$).

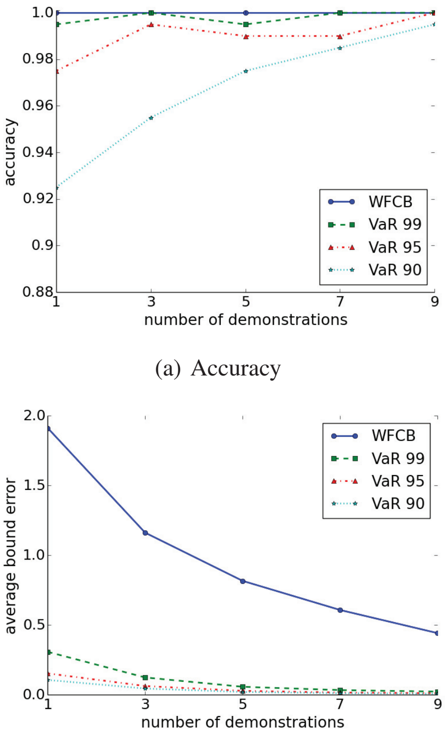
Figure 2(a) shows the accuracy of each bound where WFCB is the worst-case feature count bound, and VaR X is the X/100 quantile Value at Risk bound. The accuracy is the proportion of trials where the upper bound is greater than the ground truth expected value difference over the 200 random grid worlds. As expected, the WFCB always gives an upper bound on the true performance difference between the optimal policy and the evaluation policy. The bounds on α-VaR are also highly accurate.

Because always predicting a high upper bound will result in high accuracy, we also measured the tightness of the the upper bounds. Figure 2(b) shows the average bound error over the 200 random navigation tasks. We define the bound error for an upper bound b as

$$\text{error}(b) = b - \text{EVD}(\pi_{\text{eval}}, R^*) \qquad (10)$$

where R^* is the generated ground truth reward. We see that the bounds on the α-VaR are much tighter than the worst-case feature count bound, converging after only a small number of demonstrations.

Noisy demonstrations As mentioned previously, BIRL uses a confidence parameter, c, that represents the optimality of the demonstrations. When $c = 0$, the demonstrations are assumed to come from a completely random policy, and $c = \infty$ means that the demonstrations come from a perfectly optimal policy. Prior work used values of c between 25 and 500 when demonstrations are generated from an expert policy (Lopes, Melo, and Montesano 2009; Cohn, Durfee, and Singh 2011; Michini and How 2012b). To investigate the effect of c on our bound we generated noisy demonstrations where at step there is an 80% chance of taking an optimal action and a 20% chance of taking a random action. The resulting accuracy and bound error for several choices of c are shown in Figure 3.



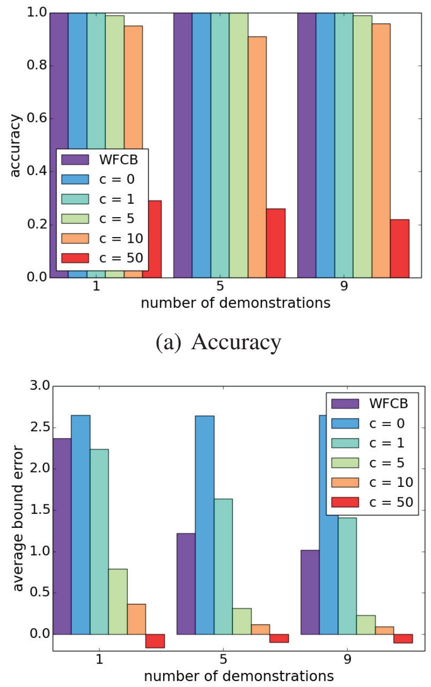
(a) Accuracy

(b) Average Bound Error

Figure 2: Results for infinite horizon grid navigation task. Accuracy and average error for bounds based on feature counts (WFCB) compared with 99, 95, and 90 percentiles for the VaR bound. Accuracy and averages are computed over 200 replicates

Adjusting c for noisy demonstrations has a clear effect on the accuracy and bound error. The bound error (Equation 10) decreases as c increases, meaning the bounds become tighter; however, when $c = 50$ the VaR bounds often underestimate the true expected value difference between the expert's policy and the evaluation policy, resulting in error$(b) < 0$ and lower accuracy. We see that values of c in the range $(1, 10]$ result in highly accuracy bounds that are tighter than the worst-case feature count bound. However, for $c = 50$, we see that BIRL overfits to the noise in the demonstrations by assuming that the demonstrations are optimal. Tuning the confidence parameter, c, for a particular demonstrator and task is left for future work.

Evaluation policy In the previous examples we have used the MAP reward obtained from BIRL to create the eval-



(a) Accuracy

(b) Average Bound Error

Figure 3: Sensitivity to the confidence c for noisy demonstrations in the grid navigation task. The demonstrator has a 20% chance of taking a random action in each state. Accuracy and average error for bounds based on feature counts (WFCB) compared with 0.95-VaR bound. Accuracy and averages are computed over 200 replicates.

uation policy; however, unlike previous theoretical confidence bounds, our method is applicable to any evaluation policy. We investigated the sensitivity of our bound over a range of different evaluation policies and found that the VaR bounds consistently outperforms the baseline WFCB, providing bounds that are often four times tighter while maintaining high accuracy (see the full paper for details (Brown and Niekum 2017)).

To demonstrate the ability of our method to work with evaluation policies derived from other IRL algorithms, and to compare against existing high-confidence bounds for IRL, we used the Projection algorithm proposed by Abbeel and Ng (Abbeel and Ng 2004) as an evaluation policy. Abbeel and Ng provide high-confidence bounds on the number of demonstrations needed for their algorithm to guarantee performance within ϵ of the demonstrator. A tighter sample bound for feature count-based methods was later derived by Syed and Schapire (2008) that also holds for the Projection algorithm. We inverted the bound of Syed and Schapire to obtain a $(1 - \delta)$ confidence bound on the expected value difference given a fixed number of demonstrations.

We then repeated the infinite horizon grid navigation experiment described above, using the policy found by the Projection algorithm as our evaluation policy. We compare the average bound error for our proposed VaR bounds with the Syed and Schapire error bound for the Projection algorithm in Table 1. Our empirical VaR bounds are two to three orders of magnitude tighter than the Hoeffding style bound which theoretically requires 23,146 demonstrations to guarantee the true EVD is within the 0.95-VaR bound found by our method using only a single demonstration.

Driving task

We now provide an example that more closely matches a real-world learning from demonstration task. Rather than evaluate our method on an ad hoc "true" reward function, we examine how the VaR bound can be used to rank and select an appropriate policy from a set of existing policies. For this task we designed a driving simulator based on previous benchmarks (Abbeel and Ng 2004; Cohn, Durfee, and Singh 2011). Figure 1(b) shows a snapshot of the simulator. The agent (blue) is in charge of driving safely down a highway and has three actions: switch lanes left, switch lanes right, or stay in current lane. The agent is traveling faster than traffic and must change lanes to avoid other cars which randomly appear at the top of the screen. There are three highway lanes where the car is supposed to drive, but it can also drive off-road on the right or left of the highway.

The state space is made up of 12 binary features: 5 features for each of the possible lanes, including the off-road lanes, 3 features telling the agent whether it is currently in collision, tailgating, or trailing another car, and 2 features for each adjacent lane, indicating whether the car will be in collision or tailgating if the car changes lanes. The reward is assumed to be a linear combination of features, $R(s) = w^T \phi(s)$, where $\phi(s)$ is a 6-dimensional binary feature vector that indicates the agent's current lane and whether it is in collision with another car. The discount factor, γ, was set to 0.9.

The goal of this experiment is to evaluate the ability of our probabilistic performance bound to correctly rank different policies, given a single demonstration of safe driving. We constructed three different evaluation policies: (1) **right-safe**: a policy that avoids hitting cars and driving off-road, but prefers driving on the right lane of the highway, (2) **on-road**: a policy that avoids driving off-road, but pays no attention to other cars, and changes lanes randomly (3) **nasty**: a policy that avoids going off-road, but actively tries to hit cars. We then generated a single demonstration of collision-free driving, consisting of 100 consecutive state-action pairs. The demonstration changed lanes randomly while avoiding collisions and avoiding driving off-road. The evaluation policies and demonstration were created using Q-learning and hand-crafted reward functions that resulted in the desired behaviors.

Because the driving task is model-free we used Q-learning to calculate the Q-values used in the likelihood calculations of BIRL. We then calculated a 95% confidence bound on the 0.95-VaR for each evaluation policy. We also computed the worst-case feature count bounds for comparison. The results are shown in Table 2.

The VaR bound uses the demonstration to focus on reward functions that are likely given the demonstrated state-action pairs. This results in correctly ranking the evaluation

	Number of demonstrations					Average Accuracy
	1	5	9	···	23,146	
0.95-VaR EVD Bound	**0.9372**	**0.2532**	**0.1328**		-	0.98
0.99-VaR EVD Bound	1.1428	0.2937	0.1535		-	1.0
EVD Bound (Syed and Schapire 2008)	142.59	63.77	47.53		0.9372	1.0

Table 1: Comparison of 95% confidence α-VaR bounds with a 95% confidence Hoeffding-style bound (Syed and Schapire 2008). Both bounds use the Projection algorithm (Abbeel and Ng 2004) to obtain the evaluation policy. Results are averaged over 200 random navigation tasks.

		Ranking (EVD upper bound)		
π_{eval}	Collisions	True	WFCB	0.95-VaR
right-safe	0	**1**	3 (5.51)	**1** (0.85)
on-road	13.65	**2**	1 (1.93)	**2** (1.09)
nasty	42.75	**3**	2 (4.11)	**3** (2.44)

Table 2: Policy rankings based on upper bounds on policy loss for three different evaluation policies in the driving domain when given a single demonstration of safe driving. Results are averaged over 20 replicates.

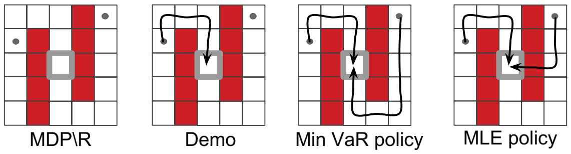


Figure 4: Given one demonstration, optimizing the VaR bound results in a risk-aware policy that hedges against the red cells being much worse than the white. The maximum likelihood reward assumes that red is only marginally worse than white.

policies. The worst-case feature count bound ignores likelihood and assumes a worst-case reward function that penalizes the largest discrepancy between the empirical feature counts of the demonstration and the expected feature counts of the evaluation policies. Because the collision feature is less frequently active than the lane features, both **on-road** and **nasty** appear safer than **right-safe** because their average state-occupancies more closely align with the state-occupancies of the demonstration.

High-confidence policy improvement

The previous section showed how we can use risk-sensitive policy evaluation to choose between multiple evaluation policies. We now take this a step further and give an example that uses risk-sensitive policy evaluation to iteratively reduce the VaR of a policy learned from demonstrations.

To highlight the potential of safe policy improvement, we consider the simple navigation task shown in Figure 4. The task has a single terminal in the center and two reward features (white and red). The agent is given a single demonstration from one starting state and must generalize this demonstration to a second starting state (both marked with circles). Note that the demonstration shows that the red feature is less desirable than the white feature, but the true magnitudes of the feature weights are left uncertain.

We implemented a simple risk-sensitive policy improvement hill climbing algorithm. We initialized the hill climbing algorithm with the maximum likelihood policy found using BIRL with a uniform prior. For each step of the hill-climbing algorithm, we examined the impact on the 0.99-VaR of changing the action taken by the policy in a single state, and chose the change that resulted in the largest decrease in 0.99-VaR over all single state changes. We continued this process until no reductions in the 0.99-VaR could be found. The resulting risk-aware policy seeks to minimize the 0.99-VaR by avoiding the red feature, whereas the maximum likelihood reward leads to a less conservative policy, resulting in a higher potential risk. The learned policies are shown in Figure 4. In the future, more complex policy adaptation schemes such as finite difference methods or black-box optimization techniques (e.g. CMA-ES (Hansen 2006)) could also be used to approximate the gradient of the α-VaR with respect to a parameterized policy π.

Related work

Many different methods exist for learning from demonstration through inverse reinforcement learning (Argall et al. 2009; Gao et al. 2012). However, few of them give guarantees on performance. Abbeel and Ng (2004) and Syed and Schapire (2008) give probabilistic Hoeffding-style bounds on how many demonstrations their algorithms require to guarantee a policy with expected return within epsilon of the expected return of the demonstrator's policy. However, as shown in Table 1, these theoretical bounds are too loose to be useful in practice and are customized for their specific IRL algorithms. To our knowledge, we provide the first sample-efficient, high-confidence bound on the policy loss of any evaluation policy with respect to the optimal policy under the demonstrator's true reward function.

Safety has been extensively studied within the reinforcement learning community (see Garcia et al. (2015) for a survey). These approaches typically either focus on safe exploration or on optimizing an objective other than expected return. Recently, alternative objectives based on financial measures of risk such as VaR and Conditional VaR have been shown to provide tractable and useful risk-sensitive measures of performance for MDPs (Tamar, Glassner, and Mannor 2015; Chow et al. 2015). Santara et al. (2017) propose an algorithm to minimize conditional VaR for generative adversarial imitation learning, but do not provide bounds on the safety of the learned policy. Our work complements prior research on safety in reinforcement learning and imitation learning by showing how risk-sensitive metrics can be applied to IRL to obtain high-confidence performance bounds.

Additional work on safety in MDPs has focused on obtaining high-confidence bounds on the performance of a policy before that policy is deployed (Thomas, Theocharous, and Ghavamzadeh 2015b; Hanna, Stone, and Niekum 2017), as well as methods for high-confidence policy improvement (Thomas, Theocharous, and Ghavamzadeh 2015a). Our work draws inspiration from these previous approaches; however, we provide bounds on policy performance that are applicable when learning from demonstrations, i.e., when the rewards are not observed.

Discussion and Future Work

Due to space and time constraints we did not explore the full range of possible instantiations of a risk-sensitive performance bound for learning from demonstration through IRL. In this section we discuss design choices, limitations of our approach, and avenues for future research.

We decided to measure policy loss using EVD as it is a commonly used IRL metric; however, this is not the only measure of performance that can be used in our approach. Because our method estimates the posterior distribution over reward functions, any risk measure or loss that is a function of a reward function and a policy can be inserted into our framework in place of EVD.

We used VaR because it well known and widely used, easy to implement using Monte-Carlo samples, and is a probabilistic analogue to the WFCB. However, our proposed methodology can be extended to use other risk measure that can be computed from samples of a distribution. Alternative risk measures such as Conditional Value-at-Risk (Rockafellar and Uryasev 2000), entropic risk measure (Föllmer and Knispel 2011), or semideviations (Ogryczak and Ruszczyński 1999) could replace VaR in our framework. Recently, methods have been proposed that explicitly optimize the Conditional VaR of a policy (Tamar, Glassner, and Mannor 2015; Santara et al. 2017). Future work should examine whether these approaches can be combined with our work on risk-aware policy improvement for IRL.

Because our bound is based on Bayesian IRL, our method is designed to work with partial demonstrations and allows insertion of domain knowledge as a prior over reward functions. Choi and Kim (2011) have shown that many standard IRL algorithms can be transformed into an equivalent Bayesian IRL algorithm by selecting the appropriate likelihood and prior. Thus, our proposed performance bound can be easily extended to use alternative likelihoods and priors that match different assumptions and preferences found in the IRL literature.

One of the main drawbacks of our proposed framework is that it requires running MCMC, which repeatedly samples rewards and then solves for $Q^*_R(s,a)$ and V^*_R in order to calculate the BIRL likelihood and to compute samples of $\text{EVD}(\pi_{\text{eval}}, R^*)$. Future work should investigate whether IRL methods based on policy gradients (Pirotta and Restelli 2016; Ho, Gupta, and Ermon 2016) or other IRL algorithms that do not require repeatedly solving an MDP (Boularias, Kober, and Peters 2011; Kalakrishnan et al. 2013; Finn, Levine, and Abbeel 2016) can be used to sample from the posterior distribution over reward functions.

Our method also relies on an appropriate range for the confidence parameter c in the BIRL algorithm, which determines how much we trust the demonstrations. Recently, an Expectation Maximization approach has been used to learn this parameter from a large number of demonstrations of differing quality (Zheng, Liu, and Ni 2014). Future work should investigate whether a similar approach can be used to learn an appropriate value for c when there are possibly only a small number of demonstrations of similar quality.

Conclusion

In this work we have formalized and addressed the problem of risk-aware high-confidence policy evaluation with an unknown reward function. To our knowledge, we present the first general framework for obtaining practical high-confidence bounds on the performance difference between an evaluation policy and the optimal policy for a demonstrator's true unknown reward. We also give examples of how our high-confidence performance bound can be used to perform risk-aware policy selection and risk-aware policy improvement. Our proposed algorithms are evaluated on a standard grid navigation task and driving simulation.

Our results demonstrate that our proposed bound is a significant improvement over a baseline based on feature counts—providing accurate, tight bounds even for small numbers of demonstrations. Additionally, our empirical results show orders of magnitude improvement in sample efficiency over competing confidence bounds (Abbeel and Ng 2004; Syed and Schapire 2008). As a result, this is the first approach that allows agents that learn from demonstrations to express confidence in the performance of their learned policy, based on limited demonstration data. We believe the techniques proposed in this paper provide a starting point for developing autonomous agents that can safely and efficiently learn from human demonstrations in risk-sensitive, real-world environments.

References

Abbeel, P., and Ng, A. Y. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the 21st international conference on Machine learning*.

Amodei, D.; Olah, C.; Steinhardt, J.; Christiano, P.; Schulman, J.; and Mané, D. 2016. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*.

Argall, B. D.; Chernova, S.; Veloso, M.; and Browning, B. 2009. A survey of robot learning from demonstration. *Robotics and autonomous systems* 57(5):469–483.

Babes, M.; Marivate, V.; Subramanian, K.; and Littman, M. L. 2011. Apprenticeship learning about multiple intentions. In *Proceedings of the 28th International Conference on Machine Learning*.

Baker, C. L.; Saxe, R.; and Tenenbaum, J. B. 2009. Action understanding as inverse planning. *Cognition* 113(3):329–349.

Boularias, A.; Kober, J.; and Peters, J. 2011. Relative entropy inverse reinforcement learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*.

Brown, D. S., and Niekum, S. 2017. Efficient probabilistic performance bounds for inverse reinforcement learning. *arXiv preprint arXiv:1707.00724* (Full paper).

Brown, D. S.; Hudack, J.; Gemelli, N.; and Banerjee, B. 2016. Exact and heuristic algorithms for risk-aware stochastic physical search. *Computational Intelligence*.

Choi, J., and Kim, K.-E. 2011. Map inference for bayesian inverse reinforcement learning. In *Advances in Neural Information Processing Systems*.

Chow, Y.; Tamar, A.; Mannor, S.; and Pavone, M. 2015. Risk-sensitive and robust decision-making: a cvar optimization approach. In *Advances in Neural Information Processing Systems*.

Cohn, R.; Durfee, E.; and Singh, S. 2011. Comparing action-query strategies in semi-autonomous agents. In *The 10th International Conference on Autonomous Agents and Multiagent Systems*.

Finn, C.; Levine, S.; and Abbeel, P. 2016. Guided cost learning: Deep inverse optimal control via policy optimization. In *International Conference on Machine Learning*.

Föllmer, H., and Knispel, T. 2011. Entropic risk measures: Coherence vs. convexity, model ambiguity and robust large deviations. *Stochastics and Dynamics* 11(02n03):333–351.

Gao, Y.; Peters, J.; Tsourdos, A.; Zhifei, S.; and Meng Joo, E. 2012. A survey of inverse reinforcement learning techniques. *International Journal of Intelligent Computing and Cybernetics* 5(3):293–311.

Garcıa, J., and Fernández, F. 2015. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research* 16(1):1437–1480.

Hanna, J. P.; Stone, P.; and Niekum, S. 2017. Bootstrapping with models: Confidence intervals for off-policy evaluation. In *Proceedings of the 16th Conference on Autonomous Agents and Multiagent Systems*.

Hansen, N. 2006. The cma evolution strategy: a comparing review. *Towards a new evolutionary computation* 75–102.

Ho, J.; Gupta, J.; and Ermon, S. 2016. Model-free imitation learning with policy optimization. In *International Conference on Machine Learning*, 2760–2769.

Hollander, M., and Wolfe, D. A. 1999. *Nonparametric Statistical Methods: By Myles Hollander, Douglas A. Wolfe*. J. Wiley.

Jorion, P. 1997. *Value at risk*. McGraw-Hill, New York.

Kalakrishnan, M.; Pastor, P.; Righetti, L.; and Schaal, S. 2013. Learning objective functions for manipulation. In *IEEE International Conference on Robotics and Automation*, 1331–1336.

Karasev, V.; Ayvaci, A.; Heisele, B.; and Soatto, S. 2016. Intent-aware long-term prediction of pedestrian motion. In *IEEE International Conference on Robotics and Automation*, 2543–2549.

Kim, B., and Pineau, J. 2016. Socially adaptive path planning in human environments using inverse reinforcement learning. *International Journal of Social Robotics* 8(1):51–66.

Levine, S.; Popovic, Z.; and Koltun, V. 2011. Nonlinear inverse reinforcement learning with gaussian processes. In *Advances in Neural Information Processing Systems*.

Lopes, M.; Melo, F. S.; and Montesano, L. 2007. Affordance-based imitation learning in robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1015–1021.

Lopes, M.; Melo, F.; and Montesano, L. 2009. Active learning for reward estimation in inverse reinforcement learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*.

Michini, B., and How, J. P. 2012a. Bayesian nonparametric inverse reinforcement learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*.

Michini, B., and How, J. P. 2012b. Improving the efficiency of bayesian inverse reinforcement learning. In *IEEE International Conference on Robotics and Automation*, 3651–3656.

Ng, A. Y., and Russell, S. J. 2000. Algorithms for inverse reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 663–670.

Ogryczak, W., and Ruszczyński, A. 1999. From stochastic dominance to mean-risk models: Semideviations as risk measures. *European Journal of Operational Research* 116(1):33–50.

Pirotta, M., and Restelli, M. 2016. Inverse reinforcement learning through policy gradient minimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 1993–1999.

Ramachandran, D., and Amir, E. 2007. Bayesian inverse reinforcement learning. In *Proceedings of the 20th International Joint Conference on Artifical intelligence*, 2586–2591.

Rockafellar, R. T., and Uryasev, S. 2000. Optimization of conditional value-at-risk. *Journal of risk* 2:21–42.

Rothkopf, C. A., and Ballard, D. H. 2013. Modular inverse reinforcement learning for visuomotor behavior. *Biological cybernetics* 107(4):477–490.

Santara, A.; Naik, A.; Ravindran, B.; Das, D.; Mudigere, D.; Avancha, S.; and Kaul, B. 2017. Rail: Risk-averse imitation learning. *arXiv preprint arXiv:1707.06658*.

Syed, U., and Schapire, R. E. 2008. A game-theoretic approach to apprenticeship learning. In *Advances in neural information processing systems*, 1449–1456.

Tamar, A.; Glassner, Y.; and Mannor, S. 2015. Optimizing the cvar via sampling. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2993–2999.

Thomas, P. S.; da Silva, B. C.; Barto, A. G.; and Brunskill, E. 2017. On ensuring that intelligent machines are well-behaved. *arXiv preprint arXiv:1708.05448*.

Thomas, P.; Theocharous, G.; and Ghavamzadeh, M. 2015a. High confidence policy improvement. In *Proceedings of the 32nd International Conference on Machine Learning*, 2380–2388.

Thomas, P. S.; Theocharous, G.; and Ghavamzadeh, M. 2015b. High-confidence off-policy evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 3000–3006.

Wulfmeier, M.; Ondruska, P.; and Posner, I. 2015. Maximum entropy deep inverse reinforcement learning. *arXiv preprint arXiv:1507.04888*.

Zheng, J.; Liu, S.; and Ni, L. M. 2014. Robust bayesian inverse reinforcement learning with sparse behavior noise. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2198–2205.

Ziebart, B. D.; Maas, A. L.; Bagnell, J. A.; and Dey, A. K. 2008. Maximum entropy inverse reinforcement learning. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, 1433–1438.