Toward Probabilistic Safety Bounds for Robot Learning from Demonstration

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

Abstract

Learning from demonstration is a popular method for teaching robots new skills. However, little work has looked at how to measure safety in the context of learning from demonstrations. We discuss three different types of safety problems that are important for robot learning from human demonstrations: (1) using demonstrations to evaluate the safety of a robot's current policy, (2) using demonstrations to enable risk-aware policy improvement, and (3) determining when the demonstrations received by the robot are sufficient to ensure a desired safety level. We propose a risk-aware Bayesian sampling approach based on inverse reinforcement learning that provides a first step towards addressing these problems. We demonstrate the validity of our approach on a simulated navigation task and discuss promising areas for future work.

Introduction

There is a growing interest in safety and risk-sensitive metrics for machine learning and artificial intelligence systems (Amodei et al. 2016), especially for systems that interact with their environment. While a growing body of work has examined safety in the context of reinforcment learning (Garcıa and Fernández 2015), little work has looked at how to measure safety for inverse reinforcement learning or learning from demonstration.

Learning from demonstration (LfD) is a popular method for teaching a robot or software agent a skill or policy by simply observing demonstrations from a human expert (Argall et al. 2009). One popular variant of LfD is Inverse Reinforcement Learning (IRL) (Ng, Russell, and others 2000) where the goal is to infer the reward function that resulted in the demonstrations. 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 safety bounds on a learned policy.

For the purposes of evaluating policies learned from demonstrations, we assume that the demonstrator is trying to optimize a unknown reward function. Thus, we define the safety of a policy as a risk-sensitive confidence bound on the difference in performance between the optimal policy for the demonstrator's reward and the performance of the robot's policy, when both policies are evaluated on the demonstrator's true reward. There are many possible methods for obtaining risk-sensitive bounds. In this work we examine both a worst-case bound based on feature-counts and a probabilistic bound based on α-Value-at-Risk (α-VaR) (Jorion 1997)—the α-quantile worst outcome. We make these definitions precise later in the paper.

In this work we propose three different types of safety problems that we believe are important for robot learning from human demonstrations:

- ***Policy evaluation through demonstrations***: using demonstrations to evaluate the safety of a robot's current policy.
- ***Policy improvement through demonstrations***: using demonstrations to improve the safety of a policy.
- ***Demonstration sufficiency***: determining when enough demonstrations have been given to ensure a desired level of safety.

We give formal definitions of these problems later; however, we first provide motivation for each problem using the example of a hospital assistant robot that is designed to lift patients out of bed.

Policy evaluation through demonstrations: As an example of the policy evaluation safety problem, consider the case where the robot comes from the factory with a pre-programmed policy designed to work well for an average patient. However, before allowing this robot to lift a patient with a particular kind of back injury, we would want to determine if the default policy is safe to use. One way to test the safety of the robot's default policy is to have an experienced human give demonstrations of lifting the patient and then have the robot evaluate the performance of its pre-programmed policy compared to the inferred policy of the demonstrator. Ideally, the robot would then be able to guarantee with high-confidence that its current lifting policy performs within some allowable error of the optimal policy under the expert's unknown reward.

Policy improvement through demonstrations: What if the robot determines that its default policy is not within an acceptable safety level for this particular patient? Ideally, we could addresses this problem by using the demonstrations provided by the expert to compute a new policy that has a

higher safety margin than the original policy, as evaluated using the solution to the policy evaluation problem.

Demonstration sufficiency: If the policy that results from policy improvement still fails the desired safety criterion, then the expert can continue to give demonstrations until the robot's confidence about its policy's performance is above a desired safety level. This motivates an iterative procedure where the robot repeatedly improves its policy by requesting more demonstrations until its policy evaluation reaches a specified level of safety.

Risk-aware approaches have been proposed and applied to physical search problems (Brown et al. 2016), planning in Markov Decision Processes (Chow et al. 2015), and reinforcement learning (Tamar, Glassner, and Mannor 2015; Garcıa and Fernández 2015). Recently, there has been interest in also applying risk-sensitive metrics to imitation learning (Santara et al. 2017); however, to the best of our knowledge, no one has investigated how to obtain sample-efficient, risk-aware safety bounds on the performance of a policy under an unknown reward function, as is the case when learning from demonstrations (Argall et al. 2009). Because the demonstrator's reward function is unknown, it may seem intractable to try and determine safety bounds that require knowing the expert's reward. Indeed, the problem of Inverse Reinforcement Learning is ill-posed—there are an infinite number of reward functions that produce in the same optimal behavior.

Thus, rather than trying to find a single reward function that explains the demonstrations, ***our key insight is to find a risk-sensitive bound on performance that takes into account the entire posterior distribution over reward functions, conditioned on the demonstrations***. To obtain this bound we use the α-Value-at-Risk (α-VaR) (Jorion 1997)—the α-quantile worst outcome—of the difference in expected return between the robot's policy and the optimal policy under the expert's true reward, when both policies are evaluated on the demonstrator's unknown reward function. We use VaR because it is much more conservative than using the expected performance of a policy, while not being as hypersensitive as an absolute worst-case bound which will focus on extreme rewards that may not match the demonstrations.

In the following sections we provide background information, make precise our definition of safety, and use this definition to formally define the three safe learning from demonstration problems described above. We then describe a Bayesian sampling approach that uses the Value at Risk cost metric from finance (Jorion 1997) that we recently proposed as a solution to the policy evaluation problem (Brown and Niekum 2017). We then build upon our previous work by providing examples of how the solution to the policy evaluation problem can be used for policy improvement and determining how many demonstrations are sufficient to learn a safe policy through demonstration.

Preliminaries

Markov decision processes

A Markov decision process (MDP) is defined as a tuple $\langle S, A, T, R, \gamma, p_0 \rangle$ 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 p_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 denoted as $V^\pi(R) = \mathbb{E}_{s_0 \sim p_0}[\sum_{t=0}^{\infty} \gamma^t R(s_t)|\pi]$. The value of executing policy π starting at state $s \in S$ is recursively defined as $V^\pi(s, R) = R(s) + \gamma \sum_{a \sim \pi(s)} \sum_{s' \in S} T(s, a, s') V^\pi(s', R)$ and the value of policy π can be written as $V^\pi(R) = \sum_{s \in S} p_0(s) V^\pi(s, R)$. Given a reward function R, the Q-value of a state-action pair (s, a) is defined as $Q^\pi(s, a, R) = R(s) + \gamma \sum_{s' \in S} T(s, a, s') V^\pi(s', R)$. We denote the optimal value and Q-value functions as $V^*(R) = \max_\pi V^\pi(R)$ and $Q^*(s, a, R) = \max_\pi Q^\pi(s, a, R)$, respectively.

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 feature weights. Thus, we can write the value of a policy as $V^\pi(R) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t w^T \phi(s_t)|\pi] = w^T \mu(\pi)$, where $\mu(\pi) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \phi(s_t)|\pi]$ 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

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

where $Q^*(s, a, R)$ 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 action taken by the expert, a, has a higher Q-value than the other alternative actions.

Samples from the posterior $P(R|D)$ are obtained through Markov Chain Monte Carlo (MCMC) sampling. 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. An estimate of the expert's reward function can be found by averaging the feature weights in

the resulting 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 rewards, (2) the state-action pairs in D can be partial demonstrations or even non-contiguous state action pairs, and (3) it allows the sub-optimality of the demonstrator to be modelled using the confidence parameter, c.

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 definitions

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 sub-optimal, stationary policy π_{demo}.

Given any evaluation policy π_{eval}, we would like to find an upper-bound on the *Expected Value Difference* (EVD) (Choi and Kim 2011) of π_{eval} under the unknown reward R^*, defined as

$$\text{EVD}(\pi_{\text{eval}}, R^*) = V^*(R^*) - V^{\pi_{\text{eval}}}(R^*) \qquad (2)$$

where $V^\pi(R) = \mathbb{E}_{s_0 \sim p_0}[\sum_{t=0}^{\infty} \gamma^t R(s_t) | \pi]$ and $V^*(R) = \max_\pi \mathbb{E}_{s_0 \sim p_0}[\sum_{t=0}^{\infty} \gamma^t R(s_t) | \pi]$. The EVD measures the difference in expected return between the evaluation policy π_{eval} and π^*, the policy that is optimal with respect to the demonstrator's unknown reward R^*.

However, because IRL is ill-posed there is an infinite family of rewards that can induce the optimal policy π^*. Because an optimal policy is invariant to any non-negative scaling of the reward function, bounding EVD is also 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 assume that $\|w\|_1 = 1$. 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 scaling problem, there can still be infinitely many rewards that induce any optimal policy.

When learning from demonstration, the main source of uncertainty is the unknown reward, and corresponding policy, of the demonstrator. Thus, to obtain a safety bound on EVD we need to address this uncertainty. As we show in the following section, one way to bound EVD is to compute a worst-case bound based on feature counts. However, as we show in the evaluation section, this type of bound is typically very loose because it is sensitive to highly unlikely adversarial reward functions. Thus, rather than focusing on absolute worst-case, we focus on computing probabilistic upper bounds on the α-worst-case EVD.

The α-worst-case value of a random variable is often referred to as the α-Value at Risk. 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

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

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

We can now formally state the three safety problems presented in the introduction:

High-confidence policy evaluation for LfD Given an MDP\R, an evaluation policy π_{eval}, and a set of demonstrations D, find a $(1 - \delta)$ confidence bound on $\nu_\alpha(\text{EVD}(\pi_{\text{eval}}, R^*))$, where R^* is the demonstrator's unobserved reward function.

High-confidence policy improvement for LfD Given an MDP\R, a baseline policy π, and a set of demonstrations D, find a new policy π' such that with $(1 - \delta)$ confidence $\nu_\alpha(\text{EVD}(\pi', R^*)) \leq \nu_\alpha(\text{EVD}(\pi, R^*))$, where R^* is the demonstrator's unobserved reward function.

High-confidence demonstration sufficiency for LfD Given an MDP\R, a learned policy π, a set of demonstrations D, and a safety margin ϵ, solve the decision problem: Is $\Pr\big(\nu_\alpha\big(\text{EVD}(\pi, R^*)\big) < \epsilon\big) \geq (1 - \delta)$, where R^* is the demonstrator's unobserved reward function?

Note that α defines the sensitivity to worst-case outliers, 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 desired risk-sensitivity.

High Confidence Policy Evaluation

The remainder of the paper is concerned with finding and evaluating solutions to the three LfD safety problems described above. We first focus on the high-confidence policy evaluation problem. In this section we derive a simple worst-case bound based on feature counts that we use as a baseline. We then present a solution to the high-confidence policy evaluation problem based on Bayesian sampling to estimate the α-VaR.

Worst-case feature count bound

This baseline is a direct extension of the the idea of using expected feature counts (Abbeel and Ng 2004) to bound the expected value difference of any evaluation policy. As a reminder, we use the notation $\mu(\pi) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \phi(s_t) | \pi]$ to represent the expected feature counts of policy π.

Given any evaluation policy π_{eval}, Abbeel and Ng (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 $V^{\pi_{\text{demo}}}(R) - V^{\pi_{\text{eval}}}(R) = w^T(\mu^* - \mu(\pi_{\text{eval}})) \leq \epsilon$ for any reward function $R(s) = w^T \phi(s)$. If π_{demo} is optimal, then $w^T(\mu^* - \mu(\pi_{\text{eval}})) = \text{EVD}(\pi_{\text{eval}}, R^*) \leq \epsilon$ so

$\|\mu^* - \mu(\pi_{\text{eval}})\|_2$ gives an upper bound on $\text{EVD}(\pi_{\text{eval}}, R^*)$. Furthermore, the worst-case feature count bound is the objective value of the following maximization problem

$$\begin{aligned} \max_w \quad & w^T(\mu^* - \mu(\pi_{\text{eval}})) && (4) \\ \text{subject to} \quad & \|w\|_1 = 1. && (5) \end{aligned}$$

This is simply an optimal resource allocation problem and the solution is to put all of our budget for w on the feature with maximal feature count difference, giving the solution $\|\mu^* - \mu(\pi_{\text{eval}})\|_\infty$.

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. Thus, the empirical *worst-case feature count bound* can be stated 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 $\hat{\mu}^*$ needs to converge to μ^*, which theoretically requires a large number of demonstrations (Abbeel and Ng 2004). Other important limitations of this bound are that it requires complete trajectories to accurately calculate expected feature counts and it does not explicitly use information about the transition dynamics of the problem or what actions were taken by the expert in which states. Finally, we note that this bound does not take into account the actual likelihood of the worst-case reward function.

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 both the structure of the problem and the likelihood of the worst-case reward function. This results in a worst-case bound that may be too loose to use in practice. Our proposed approach is to obtain a high-confidence probabilistic worst-case bound that uses the structure of the problem and the information in the demonstrations.

We propose a probabilistic confidence bound on the α-Value at Risk for $\text{EVD}(\pi_{\text{eval}}, R^*)$. Given an MDP\R and a set of state-action pairs $D = \{(s_1, a_1), \ldots, (s_m, a_m)\}$, we wish to estimate the Value at Risk of an evaluation policy π_{eval} where the α-VaR, denote by ν_α is defined as in Equation (3).

To bound the α-quantile worst-case $\text{EVD}(\pi_{\text{eval}}, R^*)$ 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 $R \sim P(R|D)$. 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 full initial state distribution. Thus, EVD measures the generalizability error of an evaluation policy by evaluating the expected return over all states with support under p_0, even if demonstrations have only been sampled from a small number of possible initial states.

Thus, computing the α-VaR of $\text{EVD}(\pi_{\text{eval}}, R)$ for $R \sim P(R|D)$ gives us an α-worst-case difference in expected return between an evaluation policy and the optimal policy for the demonstrators unknown reward function. This gives us a risk-sensitive bound that takes into account uncertainty over reward functions, while also using the structure of the MDP\R to focus on reward functions that are likely given the demonstration.

As motivated previously, we assume $\|w\|_1 = 1$ to alleviate some of the ill-posedness of the IRL problem and so we can reason over a fixed domain of weights. 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 of this algorithm can be found in (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 true reward functions given the demonstrations. For each sample $R_i \in \mathcal{R}$ 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 a sample 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, given N samples and any order statistic Y_j, we can use the normal approximation of the binomial distribution to obtain

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

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 the second line of 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. We again note that α defines the sensitivity to worst-case outliers, while $(1-\delta)$ represents our confidence in our estimate of the α-VaR.

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

input: MDP\R, π_{eval}, D, α, δ
$\mathcal{R} \leftarrow$ **BIRL**(MDP\R, D) ▷ 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 sample EVD
$Y = \text{sort}(Z)$ ▷ 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 α-VaR
return Y_k

Algorithm 2 Generic hill climbing approach to policy improvement

input: MDP\R, π_{eval}, D, α, δ, numSteps
while True do
    foundImprovement $\leftarrow$ **False**
    bestBound $\leftarrow \infty$
    for i = 1:numSteps **do**
        $\tilde{\pi} \leftarrow$ **GenerateNewPolicy**(π)
        if Algorithm 1($\tilde{\pi}$) $<$ **Algorithm 1**(π) **then**
            foundImprovement $\leftarrow$ **True**
            if Algorithm 1($\tilde{\pi}$) $<$ bestBound **then**
                bestBound $\leftarrow$ **Algorithm 1**($\tilde{\pi}$)
                $\pi_{\text{best}} \leftarrow \tilde{\pi}$
    if not foundImprovement **then**
        break
    $\pi \leftarrow \pi_{\text{best}}$
return π

Our approach introduces error by using a normal approximation; however, this error goes to zero as the number of samples increases. By the Berry-Esseen theorem (van Beek 1972), the error in the normal approximation is bounded above by $0.7655/\sqrt{N\alpha(1-\alpha)}$. Our approximation also relies on the assumption of independent samples. To ameliorate the auto-correlation in our samples obtained through MCMC, we use a skip-interval so that our bound only uses every jth sample.

The main 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, (2) it does not require optimal demonstrations, (4) it inherits from BIRL the ability to work with partial demonstrations, even disjoint state-action pairs, and (5) it allows for domain knowledge in the form of a prior.

High-confidence policy improvement

We now address the problem of how to improve an existing policy. Algorithm 1 gives us a way to compare policies based on their α worst-case expected value difference. Thus, one straightforward algorithm is to perform hill-climbing on the policy parameters π, using Algorithm 1 as a subroutine. This approach is presented in Algorithm 2.

We note that the **GenerateNewPolicy** subroutine could be as simple as randomly perturbing the policy. 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 the policy π.

Algorithm 3 Online high-confidence demonstration sufficiency

input: MDP\R, π_{eval}, α, δ, ϵ, $maxDemos$
$D \leftarrow \{d_1\}$ ▷ start with initial demonstration
if Algorithm 1(π, D) $< \epsilon$ **then**
    return True
numDemos $\leftarrow$ 1
while numDemos $<$ maxDemos **do**
    $D \leftarrow D \cup d_{\text{new}}$ ▷ get new demonstration
    numDemos = numDemos + 1
    if Algorithm 1(π, D)$< \epsilon$ **then**
        return True
return False

High-confidence demonstration sufficiency

Given a working algorithm that can solve the Policy Evaluation problem, a robot can now iterative request for additional demonstrations until it's worst-case estimate of its expected value difference is within some allowable safety tolerance level. Algorithm 3 contains pseudo-code that uses Algorithm 1 as a subroutine to solve the high-confidence demonstration sufficiency problem in an online manner, where demonstrations come one-at-a-time and the job of the robot is to signal when it has received enough demonstrations to satisfy the safety margin ϵ, provided by the human, or to report failure to reach the desired safety threshold after some maximum number of demonstrations.

Empirical results

High confidence policy evaluation

For an upper bound on the expected value difference 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 tight with respect to the true expected value difference, and (3) the previous two criteria should be true even when given a small number of demonstrations. We use a standard grid world navigation benchmark (Abbeel and Ng 2004; Ramachandran and Amir 2007; Choi and Kim 2011) to validate that our proposed VaR Bound satisfies these criteria. 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 ($\delta = 0.05$). We examine the affects of both optimal and sub-optimal demonstrations, as well as the sensitivity of our approach to the confidence parameter c and choice of evaluation policy, π_{eval}.

Grid world navigation task We empirically evaluate our approach on a standard navigation task on an $N \times N$ grid world. The 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. Each state s has a feature vector $\phi(s)$ of length F associated with it that determines the terrain type. The cost of traveling on different terrains is unknown and must be inferred from demonstrations.

To show that our results are not an artifact of a specific reward function or specific feature structure, we evaluate the performance bounds over many random grid worlds each with a randomly chosen ground truth reward. We use a 9x9 grid world navigation task with 8 one-hot binary features. We assume an initial state distribution p_0 that is uniform over 9 different states spread across the grid. When generating M demonstrations we select M states in the support of p_0, without replacement, and give a rollout from each selected initial state. When measuring accuracy and bound errors, we compare with the true expected value difference over the full initial state distribution. We used $\gamma = 0.9$ and used MCMC to generate 10,000 samples, after which we applied a burn-in of 100 and skip-interval of 20.

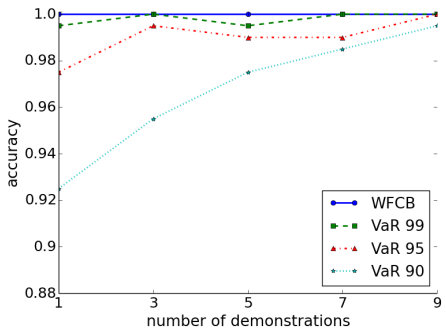
Infinite horizon grid navigation Our first task is an infinite horizon grid world navigation task with no terminal states (results for grid worlds with terminals states were very similar). To evaluate different bounding methods we generated 200 random 9x9 worlds with random features each grid cell. For each world we generated a random feature weight vector w such that $\|w\|_1 = 1$. To generate the demonstrations we solve the MDP using the generated ground truth reward to find the optimal policy. We give trajectories of length 100 for each demonstrations. We set the evaluation policy to be the optimal policy under the MAP reward function found using BIRL. Because the demonstrations are perfect, we set the BIRL confidence parameter to a large value ($c = 100$). While not the focus of our current work, in the future we believe c could also be automatically set from the demonstrations (Zheng, Liu, and Ni 2014).

Figure 1(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 calculated by counting the number of times the proposed upper bound is above the ground truth expected value difference divided by the total number of feasible rewards that were tested. Over 200 trials, 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.

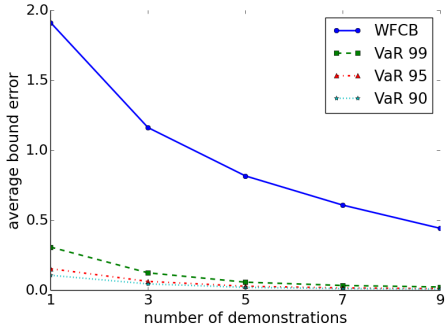
Because always predicting a high upper bound will result in high accuracy, we also measured the tightness of the the upper bounds. Figure 1(b) shows the average bound error over the 200 random navigation tasks. We measure the bound error as the difference between the upper bound and the ground truth EVD so the error for a bound b is given as

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

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.



(a) Accuracy

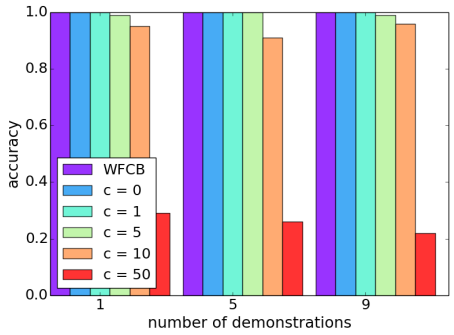


(b) Average Bound Error

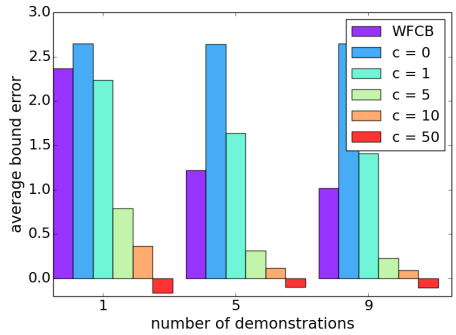
Figure 1: Results for noisy grid navigation task with no terminal states. Accuracy and average error for bounds based on feature counts (WFCB) compared with 99, 95, and 90 percentiles for the VaR bound. Accuracy and averages are computed over 200 replicates

Noisy demonstrations In real-world learning from demonstration tasks, it can be assumed that demonstrations given by a human will be noisy. As mentioned in Section , 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; Michini and How 2012). To investigate the effect of c on our results we generated demonstrations where at each demonstrated state 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 2.

Adjusting c for noisy demonstrations has a clear affect 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 experts policy and the evaluation policy, resulting in negative value of Equation 10 and lower accuracy. We see that values of c in the range $(1, 10]$ result in highly accu-



(a) Accuracy



(b) Average Bound Error

Figure 2: Sensitivity to the confidence c for noisy demonstrations in the grid navigation task. 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

racy 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.

High-confidence Policy Improvement

To highlight the potential of safe policy improvement, we consider the simple navigation task shown in Figure 3. The task has a single terminal in the center and two reward features (white and red). The robot is given a single demonstration from one starting state and must generalize this demonstration to a second starting state (both marked with circles).

We implemented the hill climbing algorithm detailed above. To generate a new policy for each step we examined the impact on the VaR of changing one state action pair in the policy and chose the change that resulted in the largest decrease in VaR for each iteration. The resulting improved policy minimizes the VaR by avoiding the red squares entirely, whereas the maximum likelihood policy finds a less conservative policy that is more likely given the demonstration, but results in a higher potential risk.

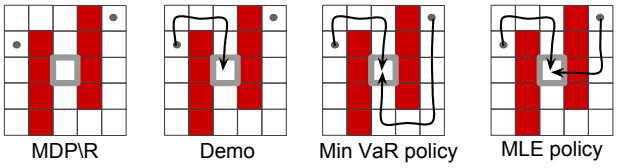


Figure 3: Given one demonstration, optimizing the VaR bound results in a safety policy that hedges against the red cells being much worse than the white. The maximum likelihood policy, simply learns that red is marginally worse than white and may result in an unsafe policy, depending on the true reward.

High-Confidence Demonstration Sufficiency

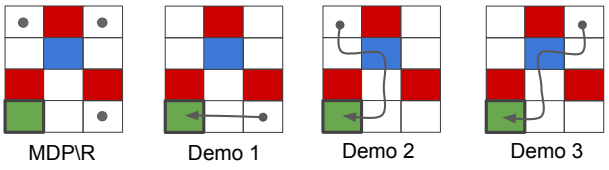
In this section we present an example that high-lights the advantages of using our VaR confidence bounds compared with the worst-case feature count bound. Figure 4 (a) shows a simple MDP\R with three features (denoted by white, blue, red, and green), one terminal state (green), and a uniform starting state distribution across the cells marked with a circle. Three demonstrations are given in the order shown. The goal of this experiment is to see when each method determines that demonstration sufficiency has been reached.

For this experiment we used the optimal policy for the MAP reward obtained by BIRL for our evaluation policy. The ground truth reward (unknown to the algorithms) was set to be a reward of +0.5 for reaching the green terminal state and -0.5 for stepping in the red states, and 0 reward everywhere else. The discount was $\gamma = 0.95$. Figure 4 (b) shows the resulting upper bounds averaged over 20 replicates of MCMC. We found that all VaR bounds were less than 0.0074 after two demonstrations. The WFCB requires three demonstrations to be confident that the learned policy is close to the optimal policy under the unknown reward function. On the other hand, the VaR methods are able to utilize knowledge of the MDP\R that the feature count bound does not use. This allows these methods to recognize that once the second demonstration is given, obtaining a third demonstration adds no information about the reward.

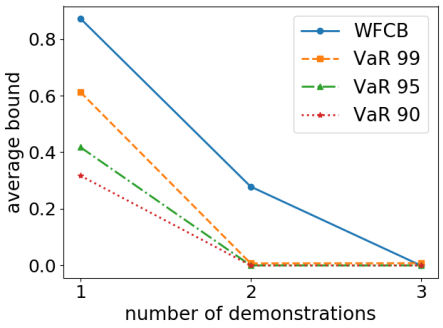
Related work

Many different methods exist for performing learning from demonstration through inverse reinforcement learning (Argall et al. 2009; Gao et al. 2012). However, few of them give any kind of sample efficient guarantees on performance. Abbeel and Ng (Abbeel and Ng 2004) give probabilistic Hoeffding-style bounds on how many demonstrations will be required to get within epsilon of the optimal policy. However, their bounds are too loose to be useful in practice and are customized for their specific IRL algorithm—we ran a brief comparison with our method and found that our bounds were three orders of magnitude more efficient in the number of demonstrations required. To our knowledge, we provide the first high-confidence performance bounds designed to work with any given evaluation policy.

Safety has been extensively studied within the reinforcement learning community (see Garcia et al. for a survey



(a)



(b)

Figure 4: (a) Shows an example MDP\R with green terminal state and starting states marked with circles. Three demonstrations are given in the order shown. (b) Shows the average EVD bound as number of demonstrations increases from one to three. Demonstration sufficiency is recognized after two demonstrations using the VaR bound over the reward posterior. The worst-case feature count bound does not converge until demonstrations are received from all initial states. Averages are computed over 20 replicates

(Garcıa and Fernández 2015)). These approaches typically either focus on safe explorationor optimize an objective other than expected long-term reward. 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 builds and extends previous work by showing how VaR can be applied to inverse reinforcement learning to enable high-confidence performance bounds.

Additional work on safety in reinforcement learning has focused on obtaining high-confidence bounds on the performance of a policy before that policy is deployed (Thomas, Theocharous, and Ghavamzadeh 2015b) as well as methods for high-confidence policy improvement (Thomas, Theocharous, and Ghavamzadeh 2015a). Unlike previous work on off-policy evaluation, we provide bounds on performance loss that are applicable when learning from demonstrations, i.e., when the rewards are not observed.

Conclusions and Future Work

In this work we have formalized and addressed the problem of high-confidence performance evaluation, when the reward function is unknown. Using this definition of safety, we then proposed three different safety problems for learning from demonstration. We then presented algorithms for solving these three problems.

Our empirical results show that our proposed VaR bound is a significant improvement over a baseline based on feature counts, and that it provides accurate, tight bounds even for small numbers of noisy demonstrations. 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. We believe that these attributes make our work an ideal starting point for developing practical safety bounds for real LfD.

Our key insight is to find a risk-sensitive bound on performance that takes into account the entire posterior distribution over reward functions, conditioned on the demonstrations. While our proposed methodology was implemented using the Value-at-Risk and the expected value difference, we believe it can easily be extended to other risk metrics.

One of the main drawbacks of our proposed VaR bound is that it requires solving an MDP at every step. 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 can be used to sample from the reward posterior. Investigating how model-free and model-based reinforcement learning algorithms can be inserted into our framework is another area of future work. Finally, our method relies on an appropriate confidence parameter c, which determines how much we trust the demonstrations. Recent work has used EM to learn this parameter from a large number of demonstrations from policies with varying amounts of noise (Zheng, Liu, and Ni 2014). Future work should investigate whether this parameter can be tuned through simple human-robot interactions.

Other related questions pertain to the choice of prior and likelihood. We used a uniform prior for our experiments. This prior was chosen to be the least biased towards any one particular reward as we hope to explore as many feasible rewards as possible to find a valid bound. Note that if you know anything a priori about the reward structure of the task, the prior provides and opportunity to inject that knowledge and would result in a better bound than using uniform. Determining which priors are best suited for the types of safety bounds we propose is an open question. Similarly, our results rely on the BIRL likelihood which is formulated as a softmax distribution over actions. We have noticed that this likelihood typically favors shaped rewards. Finding a likelihood that gives near equal weight to all rewards that give the same optimal policy would allow MCMC to sample a wider range of candidate reward functions could improve our results for problems where the true reward is sparse. Finally, running actual human subject experiments to determine what likelihood functions actually match human demonstrations is another interesting area for future work.

References

Abbeel, P., and Ng, A. Y. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, 1. ACM.

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.

Brown, D. S., and Niekum, S. 2017. Efficient Probabilistic Performance Bounds for Inverse Reinforcement Learning. *ArXiv e-prints*.

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*, 1989–1997.

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*, 1522–1530.

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.

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.

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*, 31–46. Springer.

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

Ng, A. Y.; Russell, S. J.; et al. 2000. Algorithms for inverse reinforcement learning. In *Icml*, 663–670.

Pirotta, M., and Restelli, M. 2016. Inverse reinforcement learning through policy gradient minimization. In *AAAI*, 1993–1999.

Ramachandran, D., and Amir, E. 2007. Bayesian inverse reinforcement learning. *Urbana* 51(61801):1–4.

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*.

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. AAAI Press.

Thomas, P.; Theocharous, G.; and Ghavamzadeh, M. 2015a. High confidence policy improvement. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2380–2388.

Thomas, P. S.; Theocharous, G.; and Ghavamzadeh, M. 2015b. High-confidence off-policy evaluation. In *AAAI*, 3000–3006.

van Beek, P. 1972. An application of fourier methods to the problem of sharpening the berry-esseen inequality. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* 23(3):187–196.

Zheng, J.; Liu, S.; and Ni, L. M. 2014. Robust bayesian inverse reinforcement learning with sparse behavior noise. In *AAAI*, 2198–2205.

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