4th Symposium on Advances in Approximate Bayesian Inference, 2022 1–8

Efficient Bayesian Inverse Reinforcement Learning via Conditional Kernel Density Estimation

Anonymous Authors
Anonymous Institution

Abstract

Inverse reinforcement learning (IRL) methods attempt to recover the reward function of an agent by observing its behavior. Given the large amount of uncertainty in the underlying reward function, it is often useful to model this function probabilistically, rather than estimate a single reward function. However, existing Bayesian approaches to IRL use a Q-value function to approximate the likelihood, leading to a computationally intractable and inflexible framework. Here, we introduce kernel density Bayesian IRL (KD-BIRL), a method that uses kernel density estimation to approximate the likelihood. This lends itself to an efficient posterior inference for the reward function given a sequence of agent observations. Using both linear and nonlinear reward functions in a Gridworld environment, we demonstrate that the KD-BIRL posterior centers around the true reward function, and that our method is more efficient than existing approaches.

1. Introduction

Reinforcement learning (RL) methods find policies and sequences of actions that maximize an agent's long-term expected reward in a Markov decision process (MDP). However, in many off-policy, observational data settings, we observe a sequence of states and actions for an agent who is carrying out a policy based on a reward function that is unknown to the observer. In these cases, it is of interest to infer the reward function that the agent is using in order to understand the factors driving certain behavioral patterns. For example, in a hospital setting, we may observe the treatment schedule a patient received along with measurements of the patient's health state, which can be represented as physiological covariates and clinical interventions. In this case, we may be interested in understanding the doctor's reward function—which is typically complex and mostly unobserved—and how this function drives certain treatment decisions based on a patient's state.

For this purpose, inverse reinforcement learning (IRL) methods aim to recover the reward function (i.e., objectives or priorities) of an agent given observations of the agent's behavior. The earliest IRL algorithms focused on estimating a single reward function (a point estimate) that best explained the observed behavior (Abbeel and Ng, 2004; Ng and Russell, 2000). These IRL frameworks led to applications in path planning (Mombaur et al., 2010), urban navigation (urb), and robotics settings such as quadruped locomotion (Ratliff et al., 2006a; Kolter et al., 2008). IRL is also broadly useful to generate better RL policies; if we can infer the reward function from a set of desired behavior demonstrations, we can then fit new policies using this reward function, which may lead to better outcomes.

Despite the success of these early IRL approaches, there are major limitations to estimating a single reward function, and it is often impossible to recover the exact reward function of an agent for a number of reasons. First, the IRL problem is likely non-identifiable (urb;

© A. Authors.

Ziebart et al., 2009; Ratliff et al., 2006b; Abbeel and Ng, 2004). That is, there may be many reward functions that explain a set of behaviors in an agent equally well. Second, if the training data are insufficient to identify the reward function, point estimates generally fail to capture the uncertainty in these estimates. Thus, it is advantageous to take a Bayesian approach and express our uncertainty through a posterior distribution over the estimated reward function (Ramachandran and Amir, 2007; Balakrishnan et al., 2020; Chan and van der Schaar, 2021; Michini and How, 2012a,b; Choi and Kim, 2012). A Bayesian approach communicates a degree of confidence that relies on the dataset distribution, and the posterior will ideally place mass on all regions of the reward function space that could explain the observed behavior.

Despite advances in this area, existing Bayesian IRL methods tend to be computationally demanding. This is because, in sampling-based inference for the posterior, each reward function must be evaluated using forward RL by fitting a policy using that sampled function. A single instance of forward RL is already computationally demanding, so requiring it for each sampled reward function compounds the problem.

To address this issue, we introduce kernel density Bayesian inverse reinforcement learning (KD-BIRL), an efficient IRL framework that eliminates the need to fit a new policy for each sampled reward function. Our contributions to the IRL literature are as follows: (1) KD-BIRL calculates the likelihood using a conditional kernel density estimation that does not require forward RL to be run at inference time; (2) We show that KD-BIRL is far more computationally feasible than related approaches; (3) We demonstrate that KD-BIRL returns accurate posterior estimates with both linear and nonlinear reward functions.

2. Related Work

In the first paper to propose Bayesian IRL (Ramachandran and Amir, 2007), the authors use a Q-value function to calculate the likelihood of seeing a given state and action pair (s, a) given a reward function R. This method is generally unsuitable for complex state spaces because it requires us to repeatedly solve an environment's MDP using forward RL. In particular, the algorithm uses Markov chain Monte Carlo (MCMC) sampling to compute the Gibbs posterior; every iteration requires a likelihood calculation, and each of these calculations require that forward RL is run to refit the Q-value function to the current reward function. This is computationally prohibitive, especially with infinite or real-valued state spaces.

To address this issue, it is necessary to either formulate a different likelihood estimator or minimize the number of calls to forward RL. Recent work proposes a new likelihood calculation that relies on human-recorded pairwise preferences over the demonstrations (Brown and Niekum, 2019). While this greatly reduces the computational complexity of the earlier methods, it relies on having these preferences available. The preferences may not be known, especially when the state spaces are large and the number available demonstrations are small. Another approach samples more appropriate reward functions using variational inference (Chan and van der Schaar, 2021); this approach requires far fewer calls to forward RL, making it more appropriate for high-dimensional environments. However, it still relies on a Q-value function estimate to compute the observed data likelihood, and this is computationally expensive. The use of a Q-value function implies that forward RL must

be run every time the likelihood of a reward function is evaluated, which is time consuming for even moderately complex environments.

3. Kernel density Bayesian inverse reinforcement learning

In this section we present the kernel density Bayesian inverse reinforcement learning (KD-BIRL) algorithm. We first discuss the setup of the IRL problem, and then describe the likelihood calculation and how to estimate the reward function posterior distribution.

The input to an IRL algorithm is a set of expert demonstrations of an optimal policy for a fixed but unspecified reward function. Given these demonstrations, the goal is to recover the real-valued reward function $R : \mathcal{S} \to \mathbb{R}$, where $\mathcal{S}$ is the state space. Each sample in a demonstration at time i is a 2-tuple (s_i, a_i) representing the agent's state and chosen action. Note that rewards are not available explicitly in these demonstrations. Given demonstrations of an optimal policy with respect to an unknown reward function $\pi^\star : \mathcal{S} \to \mathcal{A}$, where $\mathcal{A}$ is the action space, IRL seeks to characterize that reward function R such that $\pi^\star$ is optimal. We can derive $\pi^\star$ as follows.

Suppose that $\mathcal{S} \subseteq \mathbb{R}^p$, and let $s \in \mathcal{S}$ denote a state vector. Let $\mathcal{R}$ denote the space of reward functions. The value function for a given policy π is $V^\pi = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t)|b_0, \pi]$ where b_0 is the probability of starting in state s_t. Without loss of generality, assume that the optimal policy is given by $\pi^\star(s) = a^\star$, where $a^\star$ is the optimal action to be taken at state s. In order to identify a reward function R for which π is optimal, it must be true that $\mathbb{E}_{s' \sim P_{sa^\star}} [V^\pi(s')] \geq \mathbb{E}_{s' \sim P_{sa}} [V^\pi(s')]$, for all states $s \in \mathcal{S}$ and all actions $a \in \mathcal{A} \setminus a^\star$, where P_{sa} is the state transition probability. In other words, the long-term expected value of the optimal policy under the reward must be equal to or greater than the long-term expected value of selecting an action that deviates from the optimal policy.

Generalizing beyond linear functions, our goal here is to infer a posterior distribution over an expert reward function R from n demonstrations of the expert policy $\{(s_i^e, a_i^e)\}_{i=1}^n$. The posterior density is proportional to the product of the prior distribution $p(R)$ on the rewards and the likelihood of the demonstrations given the reward:

$$p\left(R \mid \{(s_i^e, a_i^e)\}_{i=1}^n\right) \propto p(R) \prod_{i=1}^{n} p(s_i^e, a_i^e | R). \tag{1}$$

In the original Bayesian IRL framework (Ramachandran and Amir, 2007), the form of the "likelihood" is

$$p(s_i, a_i \,|\, R) \propto e^{\alpha Q^\star(s_i, a_i, R)}, \tag{2}$$

where $Q^\star(s_i, a_i, R)$ is the optimal Q-value function computed with reward R and $\alpha > 0$ is a temperature parameter. In fact, Equation (2) is a loss-based function instead of a likelihood, and the resulting posterior is the Gibbs posterior instead of a true posterior. Additionally, the likelihood above depends on the Q-value function, which is expensive to recompute for each estimate of the reward function.

We propose a simple nonparametric method, a conditional kernel density estimator (CKDE) based on Gaussian kernels. CKDE has been well-studied, with known asymptotic behaviors (van der Vaart, 2000). This calculation requires m demonstration samples from agents that follow different reward functions.The training dataset consists of 3-tuples

$\{(s_j, a_j, R_j)\}_{j=1}^m$, where R_j is the reward functions generating (s_j, a_j). Given these samples, the conditional density given the reward R is

$$\widehat{p}_m(s, a \mid R) \propto \sum_{j=1}^{m} \frac{e^{-d_s((s,a),(s_j,a_j))^2/(2h)} e^{-d_r(R,R_j)^2/(2h')}}{\sum_{\ell=1}^{m} e^{-d_r(R,R_\ell)^2/(2h')}}, \tag{3}$$

where $d_s : (\mathcal{S}\times\mathcal{A})\times(\mathcal{S}\times\mathcal{A}) \to \mathbb{R}$ is a distance metric to compare (s, a) tuples, $d_r : \mathcal{R}\times\mathcal{R} \to \mathbb{R}$ is a distance metric to compare reward functions, and $h, h' > 0$ are smoothing hyperparameters. We can now estimate $p(s_i^e, a_i^e|R)$ by $\widehat{p}_n(s_i^e, a_i^e|R)$, where (s_i^e, a_i^e) are samples in the expert trajectories:

$$\widehat{p}_m(s_i^e, a_i^e|R) \propto \sum_{j=1}^{m} \frac{e^{-d_s((s_i^e,a_i^e),(s_j,a_j))^2/(2h)} e^{-d_r(R,R_j)^2/(2h')}}{\sum_{\ell=1}^{m} e^{-d_r(R,R_\ell)^2/(2h')}}. \tag{4}$$

The estimated posterior is then calculated as

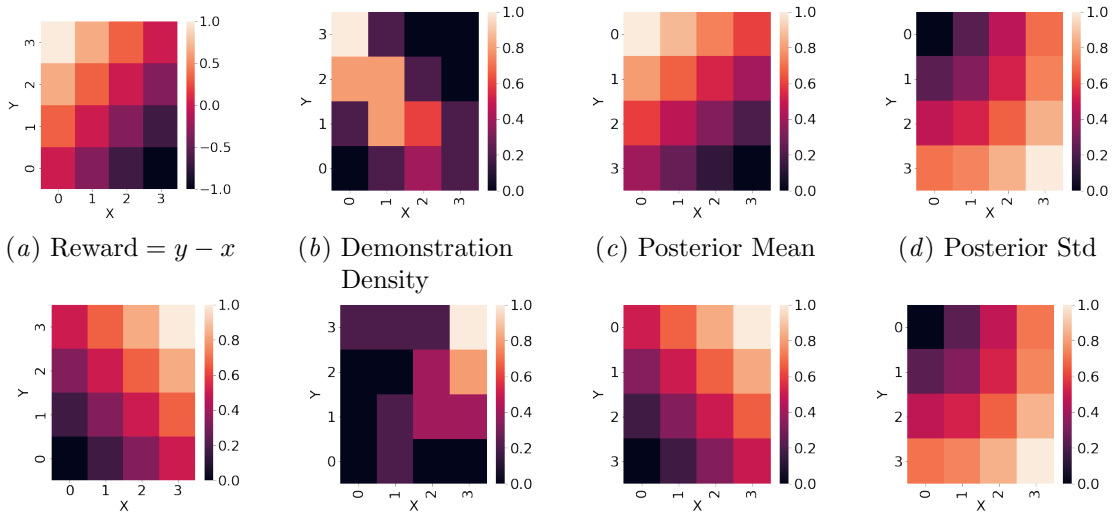
$$\widehat{p}_m(R|\{s_i^e, a_i^e\}_{i=1}^n) \propto p(R) \prod_{i=1}^{n} \sum_{j=1}^{m} \frac{e^{-d_s((s_i^e,a_i^e),(s_j,a_j))^2/(2h)} e^{-d_r(R,R_j)^2/(2h')}}{\sum_{\ell=1}^{m} e^{-d_r(R,R_\ell)^2/(2h')}}. \tag{5}$$

Note that once the conditional density $\widehat{p}_m$ is obtained, it is not necessary to run forward RL to evaluate the posterior at any reward R. This eliminates the vast majority of the computational complexity of existing BIRL algorithms and allows us to easily apply this algorithm to complex environments with large state spaces. Additionally, as $m \to \infty$, $\widehat{p}_m$ converges to the true conditional density and the posterior converges to the true posterior (van der Vaart, 2000).

4. Experiments

In this section, we perform a series of experiments to test the accuracy and robustness of this approach and compare it to AVRIL (Chan and van der Schaar, 2021), a competing method. We perform these experiments in a modified Gridworld environment, with uniform prior on the reward function. Gridworld has a two-dimensional discrete state space that lies on an $\ell \times \ell$ grid. Specifically, the MDP is: $\mathcal{S} = [\ell] \times [\ell]$, $\mathcal{A} = \{[0, 0], [0, 1], [1, 0], [-1, 0], [0, -1]\}$, $R : \mathcal{S} \to \mathbb{R}$, where we choose a grid size of $\ell = 4$, and the actions correspond to not moving ($[0, 0]$), going up ($[0, 1]$), right ($[1, 0]$), left ($[-1, 0]$), and down ($[0, -1]$). Each grid contains a target state, and the reward depends on the coordinates of the state, with the maximum reward received at the target state. Our method requires that we define two distance metrics: one associated with the distance between reward functions (d_r), and one associated with the distance between state-action pairs (d_s). In this setting, for d_r, we use the cosine distance for linear rewards and the Euclidean distance for nonlinear rewards, and for d_s, we use the Euclidean distance. We choose $n = 30$, $m = 1000$, $h = 0.03$, $h' = 0.03$, and k is between 4 and 8.

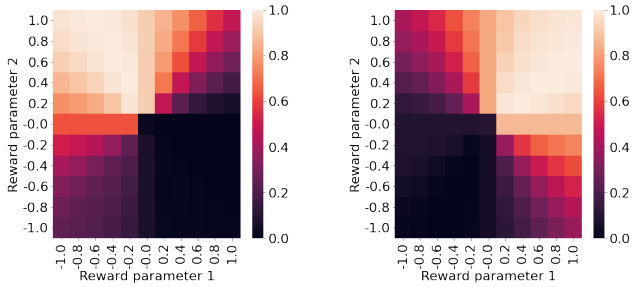
We first validate that our algorithm is able to recover the posterior distribution of a simple linear reward function $R(s) = \beta^\top s$, where $s \in \mathcal{S}$ and $\beta \in \mathbb{R}^2$. We show results for two sets of linear reward function parameters: $\beta = [-1, 1]$ and $\beta = [1, 1]$ (Figure 1). In these results, we also visualize the demonstration density, a measure of the relative state



(a) Reward $= y - x$ (b) Demonstration Density (c) Posterior Mean (d) Posterior Std

(e) Reward $= x + y$ (f) Demonstration Density (g) Posterior Mean (h) Posterior Std

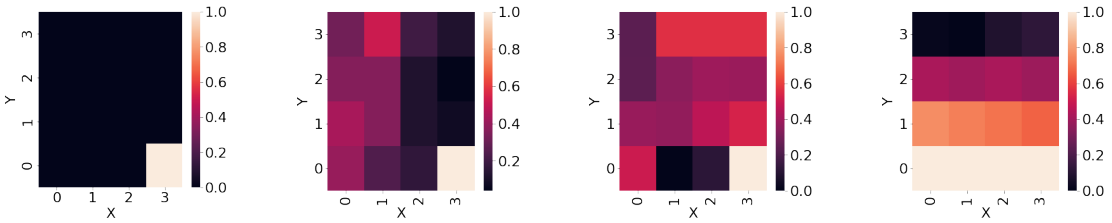
Figure 1: Linear reward functions and learned results using KD-BIRL on a Gridworld map. KD-BIRL returns an accurate estimate on a linear function with parameters $[-1, 1]$ (top row) and $[1, 1]$ (bottom row). We also visualize the associated demonstration density (Panel b, f), posterior estimate mean (Panel c, g), and standard deviation (Panel d, h).



(a) $\beta = (-1, 1)$ (b) $\beta = (1, 1)$

Figure 2: Visualization of the expert posterior distributions on a grid of reward parameters $\beta = [-1, 1]$ (Panel a) and $\beta = [1, 1]$ (Panel b). The KD-BIRL posteriors are the most confident about parameters that reflect the same signs and relative magnitude.

occupancy of the expert demonstrations. We also visualize the posterior distribution by evaluating it on a grid of reward parameters to understand how uncertain our estimates are. We find that KD-BIRL is confident about reward parameters that most effectively characterize the expert demonstrations, and its posterior centers around the true reward function (Figure 2). Finally, we consider non-linear reward functions. Here, we use a



(a) Reward $= \delta_{(3,0)}$ (b) Demonstration density (c) KD-BIRL (d) AVRIL

Figure 3: Comparison of KD-BIRL and AVRIL for a nonlinear reward function. The scaled mean reward estimate of KD-BIRL (Panel e) recovers a reward estimate that better reflects the ground truth than mean reward estimate of AVRIL (Panel d).

function where the reward is 0 in all states except the target state, which receives a reward of 1 (Figure 3). We also compare our estimate to that of AVRIL (Chan and van der Schaar, 2021). Our experiments suggest that KD-BIRL can recover posterior estimates over linear and non-linear reward functions in a Gridworld environment. We find that while the AVRIL algorithm is more computationally efficient than earlier approaches, it can still return inaccurate posterior estimates (Figure 3).

5. Discussion and future directions

In this work, we introduce a novel approach to estimate conditional densities for the purpose of inverse reinforcement learning. We demonstrate that this approach to Bayesian IRL removes much of the computational complexity associated with similar algorithms and does not need to solve an MDP using forward RL to calculate the likelihood of a given set of reward parameters.

Several future directions remain. This work is currently adapted best for on-policy (i.e., simulation) environments, and we believe that additional work must be done to apply it on off-policy environments such as clinical settings. The particular choice of distance metrics used in the conditional density estimate is also likely to depend on the environment and reward function type; we pick metrics that are most suitable to the Gridworld environment in this paper, and we believe that additional experimentation must be done to adapt this to different environments. The same is true for the hyperparameters h and h'; we currently use rule-of-thumb hyperparameters (Silverman, 1986), but additional work is needed to determine if this is reasonable. Furthermore, our current calculation of the conditional density is most suitable for environments with a low-dimensional state space. To extend this to a higher dimensional state space, like in clinical environments, we anticipate using a Gaussian process could be beneficial. Finally, additional work must be done to develop methods to evaluate this algorithm in off-policy environments, where the true reward function is unknown.

References

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

Sreejith Balakrishnan, Quoc Phong Nguyen, Bryan Kian Hsiang Low, and Harold Soh. Efficient exploration of reward functions in inverse reinforcement learning via bayesian optimization. *arXiv preprint arXiv:2011.08541*, 2020.

Daniel S. Brown and Scott Niekum. Deep Bayesian reward learning from preferences. *ArXiv*, abs/1912.04472, 2019.

Alex James Chan and Mihaela van der Schaar. Scalable Bayesian inverse reinforcement learning. In *International Conference on Learning Representations*, 2021.

Jaedeug Choi and Kee-eung Kim. Nonparametric Bayesian inverse reinforcement learning for multiple reward functions. In *Advances in Neural Information Processing Systems*, volume 25, 2012.

J Zico Kolter, Pieter Abbeel, and Andrew Y Ng. Hierarchical apprenticeship learning with application to quadruped locomotion. In *Advances in Neural Information Processing Systems*, pages 769–776. Citeseer, 2008.

Bernard Michini and Jonathan P. How. Bayesian nonparametric inverse reinforcement learning. In Peter A. Flach, Tijl De Bie, and Nello Cristianini, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 148–163, Berlin, Heidelberg, 2012a. Springer Berlin Heidelberg.

Bernard Michini and Jonathan P. How. Improving the efficiency of Bayesian inverse reinforcement learning. In *2012 IEEE International Conference on Robotics and Automation*, pages 3651–3656, 2012b.

Katja Mombaur, Anh Truong, and Jean-Paul Laumond. From human to humanoid locomotion—an inverse optimal control approach. *Autonomous Robots*, 28(3):369–383, 2010.

Andrew Y. Ng and Stuart Russell. Algorithms for inverse reinforcement learning. In *in Proc. 17th International Conf. on Machine Learning*, pages 663–670. Morgan Kaufmann Publishers Inc., 2000.

Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcement learning. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, page 2586–2591, 2007.

Nathan Ratliff, David Bradley, J. Andrew Bagnell, and Joel Chestnutt. Boosting structured prediction for imitation learning. In *Advances in Neural Information Processing Systems*, page 1153–1160, 2006a.

Nathan D. Ratliff, J. Andrew Bagnell, and Martin A. Zinkevich. Maximum margin planning. 2006b.

B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London, 1986.

Aad van der Vaart. *Asymptotic Statistics*, volume 3. Cambridge University Press, 2000.

Brian D. Ziebart, Andrew L. Maas, J. Andrew Bagnell, and Anind K. Dey. Human behavior modeling with maximum entropy inverse optimal control. In *AAAI Spring Symposium: Human Behavior Modeling*, 2009.