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

Efficient Bayesian Inverse Reinforcement Learning via Conditional Kernel Density Estimation

Aishwarya Mandyam AISHWARYA@PRINCETON.EDU
Princeton University, Gladstone Institutes

Didong Li DIDONGLI@PRINCETON.EDU
Princeton University

Diana Cai DCAI@CS.PRINCETON.EDU
Princeton University

Andrew Jones AJ13@PRINCETON.EDU
Princeton University

Barbara E. Engelhardt BEE@CS.PRINCETON.EDU
Princeton University, Gladstone Institutes

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, or the probability of the observed states and actions given a reward function. This approximation allows for efficient posterior inference of the reward function given a sequence of agent observations. Empirically, using both linear and nonlinear reward functions in a Gridworld environment, we demonstrate that the KD-BIRL posterior centers around the true reward function.

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 for a patient along with measurements of the patient's health state, as represented by 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. Early 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 (Ziebart et al., 2008), and robotics settings, such as quadruped locomotion (Ratliff et al., 2006a; Kolter et al., 2008). Additionally, a challenge in RL is designing reward functions for achieving desired behavior patterns; IRL can be used in this setting to infer the reward function from a set of desired behavior demonstrations, which can then be used to fit improved RL policies.

Despite the success of early IRL approaches, there are major limitations to point estimation in this setting. First, the IRL problem is often non-identifiable (Ziebart et al., 2008, 2009; Ratliff et al., 2006b; Abbeel and Ng, 2004). That is, there are multiple (and possibly infinite) reward functions that explain a set of behaviors in an agent equally well. Second, for finite training datasets, point estimates fail to capture the uncertainty and noise in the data generating process. Thus, it is advantageous to take a Bayesian approach and express 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, placing mass on all regions of the reward function space that could explain the observed behavior.

Despite advances in Bayesian IRL, existing methods are computationally demanding. This is because, in sampling-based inference for the posterior, these methods require value iteration for each instance of a reward function. A single instance of value iteration is in itself computationally demanding, so repeating 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) We introduce KD-BIRL, an IRL method that calculates the likelihood using a conditional kernel density estimation that does not require value iteration at inference time; (2) We show that KD-BIRL is 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

The first work exploring Bayesian IRL (Ramachandran and Amir, 2007) used 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 unsuitable for complex state spaces because it requires us to repeatedly solve an environment's MDP using value iteration to learn a Q-value function. The proposed inference algorithm uses Markov chain Monte Carlo (MCMC) sampling to compute a Gibbs posterior. Every sampling iteration requires running value iteration to estimate the Q-value function for the sampled reward function. This step 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 times value iteration is performed. Recently, a likelihood approximation was proposed that relies on human-recorded pairwise preferences over the

demonstrations (Brown and Niekum, 2019). While this greatly reduces the computational complexity of earlier methods, these human-recorded preferences are not available in most settings. Another line of work has proposed approximating the exact reward function posterior using variational inference (Chan and van der Schaar, 2021); this approach requires far fewer instances of value iteration, making it more appropriate for high-dimensional environments. However, it still relies on an expensive Q-value function estimate to compute the likelihood of the observed data.

3. Background: Inverse reinforcement learning

In the IRL setting, the data we wish to model are a set of expert demonstrations, $\{(s_i, a_i)\}_{i=1}^n$, where the demonstration at time i is a 2-tuple (s_i, a_i) representing the agent's state and chosen action. These demonstrations are assumed to arise from an agent acting according to an optimal policy, $\pi^\star : \mathcal{S} \to \mathcal{A}$, for a fixed but unknown reward function $R : \mathcal{S} \to \mathbb{R}$, where $\mathcal{A}$ is the action space and $\mathcal{S}$ is the state space. Given these demonstrations, the IRL objective is to recover R. Specifically, IRL seeks the set of reward functions such that $\pi^\star$ is optimal.

We now formalize this problem. 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, and $\gamma \in (0, 1)$ is a discount factor. 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 $P_{sa} = p(s'|s, a)$ 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.

Taking a Bayesian approach to IRL, our goal is to infer the posterior distribution over the reward function R given n demonstrations of the expert policy $\{(s_i^e, a_i^e)\}_{i=1}^n$. We call this dataset the set of *expert demonstrations*. The posterior density is proportional to the product of the prior distribution on the rewards, $p(R)$, and the likelihood of the expert 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

$$p(s, a \mid R) \propto e^{\alpha Q^{\star(s,a,R)}}, \tag{2}$$

where $Q^\star(s_i, a_i, R)$ is the optimal Q-value function computed with reward R and $\alpha > 0$ is an inverse temperature parameter. Equation (2) is a loss-based function instead of a likelihood, and the resulting posterior is the Gibbs posterior instead of a true posterior (Shawe-Taylor and Williamson, 1997). Additionally, the likelihood above depends on the Q-value function, which is expensive to recompute for each sampled reward function.

4. Kernel density Bayesian inverse reinforcement learning

We propose kernel density Bayesian inverse reinforcement learning (KD-BIRL), which uses a nonparametric conditional kernel density estimator (CKDE) with Gaussian kernels to estimate the likelihood in the IRL posterior (Equation 1). CKDEs have been well-studied, with known asymptotic behaviors (van der Vaart, 2000).

To fit the KDE, we propose sampling another set of demonstrations to augment the observed expert demonstrations. Specifically, we sample m demonstrations from agents whose behaviors are optimal for reward functions distinct from the expert reward function. This dataset, which we call 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). Using the training dataset, the conditional density for a state-action pair (s, a) given a reward function R is

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

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 the likelihood $p(s_i^e, a_i^e|R)$ by $\widehat{p}_n(s_i^e, a_i^e|R)$, where (s_i^e, a_i^e) is a single sample from the dataset of expert demonstrations:

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

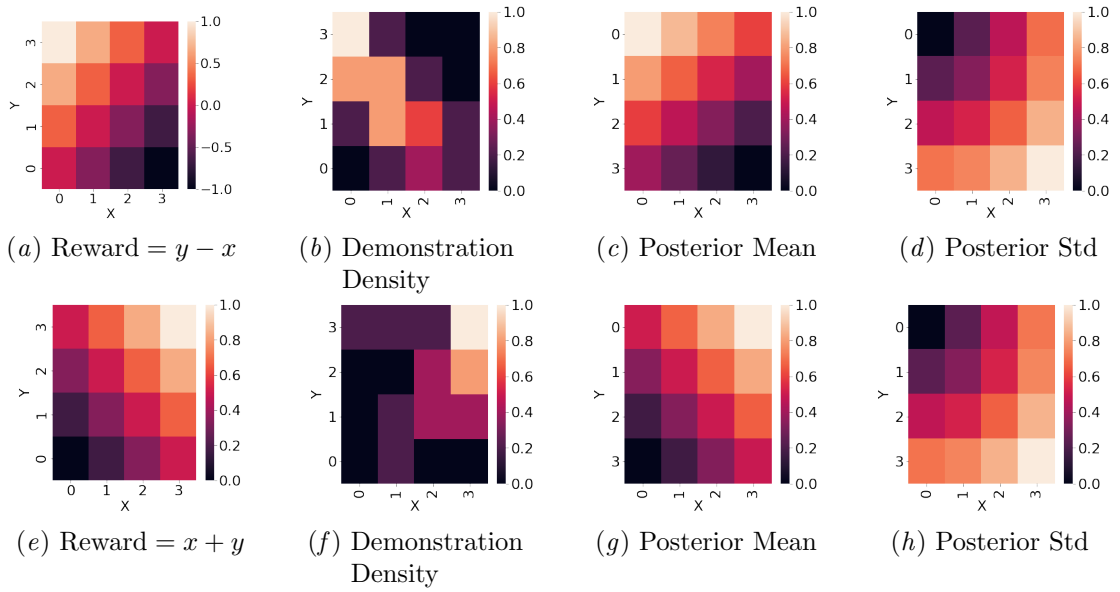
The estimated posterior is then

$$\widehat{p}_m(R|\{s_i^e, a_i^e\}_{i=1}^n) \propto p(R) \prod_{i=1}^{n} \sum_{j=1}^{m} \frac{e^{-d_s((s_i^e,a_i^e),(s_j,a_j))^2/(2h)} e^{-d_r(R,R_j)^2/(2h')}}{\sum_{\ell=1}^{m} e^{-d_r(R,R_\ell)^2/(2h')}}. \tag{3}$$

Importantly, note that once the conditional density $\widehat{p}_m$ is obtained, it is not necessary to perform value iteration to evaluate the posterior for a given reward function R. This drastically reduces the computational complexity compared to existing BIRL algorithms and opens the door for performing Bayesian IRL in complex environments with large state spaces. Additionally, as $m \to \infty$, $\widehat{p}_m$ converges to the true likelihood and the posterior converges to the true posterior (van der Vaart, 2000).

5. Experiments

In this section, we perform a series of experiments to test the accuracy and robustness of KD-BIRL and compare it to AVRIL (Chan and van der Schaar, 2021), a competing method. We perform these experiments in a modified Gridworld environment with a linear reward function, and we specify a uniform prior on the reward function parameters. Gridworld has a two-dimensional discrete state space made up of tiles on an $\ell \times \ell$ grid. 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 ($[0, 0]$), moving up ($[0, 1]$), moving right ($[1, 0]$), moving left ($[-1, 0]$), and moving down ($[0, -1]$). A grid contains a target state (or



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

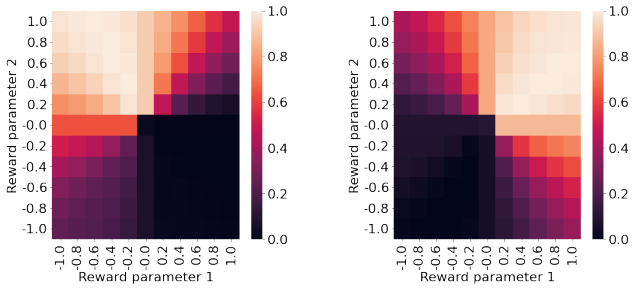
(e) Reward $= x + y$ (f) Demonstration Density (g) Posterior Mean (h) Posterior Std

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).

goal state). The reward observed in a state depends on the coordinates of the state, with the maximum reward received at the target state. KD-BIRL 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 ℓ is between 4 and 8.

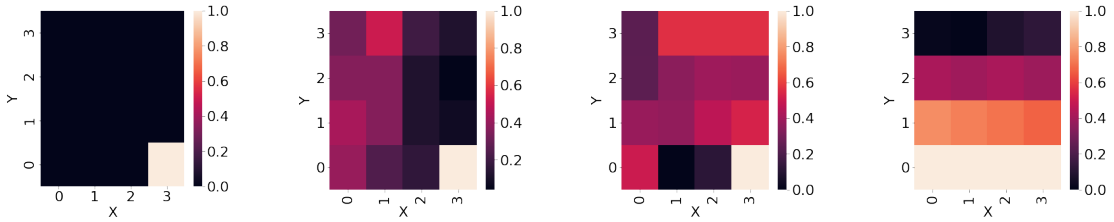
We first validate that our algorithm is able to recover the posterior distribution of a simple linear reward function $R(s) = \beta^\top s$, where $\beta \in \mathbb{R}^2$ is a coefficient vector. We fit KD-BIRL for expert demonstrations generated from two linear reward functions: $\beta = [-1, 1]$ and $\beta = [1, 1]$ (Figure 1). In these results, we also visualize the demonstration density, a measure of the relative state occupancy of the expert demonstrations. We also visualize the posterior distribution over reward functions by evaluating the posterior (Equation 3) on a grid of reward parameters to understand how uncertain our estimates are, and whether their mass tends to be situated around the functions used to generate the data. 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 function where the reward is 0 in all states except the target state, which receives a reward of 1 (Figure 3). In this experiment, we specify the set of possible reward functions to be fully nonparametic,



(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.



(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 c) recovers a reward estimate that better reflects the ground truth than mean reward estimate of AVRIL (Panel d).

and allow each of the 16 possible states to have its own estimated scalar reward. We also compare our estimate to that of AVRIL (Chan and van der Schaar, 2021). We find that KD-BIRL's posterior has a sharp peak at the target state, and much lower mass elsewhere. Collectively, 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).

6. Discussion and future directions

In this work, we introduce KD-BIRL, a novel approach to inverse reinforcement learning using conditional kernel density estimators. We demonstrate that this approach to Bayesian IRL removes much of the computational complexity associated with similar algorithms and does not need to repeatedly solve an MDP using value iteration to calculate the likelihood of a given set of reward parameters.

Several future directions remain. This work is currently well-suited 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 medical or clinical decision making 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 could further optimize these values. 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.

7. Acknowledgements

This work was funded by the Helmsley Trust grant AWD1006624, NIH NCI 5U2CCA233195, NIH NHLBI R01 HL133218, and NSF CAREER AWD1005627. BEE is on the SAB of Creyon Bio, Arrepath, and Freenome. D. Cai was supported in part by a Google Ph.D. Fellowship in Machine Learning.

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.

John Shawe-Taylor and Robert C Williamson. A pac analysis of a bayesian estimator. In *Proceedings of the tenth annual conference on Computational learning theory*, pages 2–9, 1997.

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, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

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.