Bayesian Inverse Reinforcement Learning for Expert's Demonstrations in Multiple Dynamics

Yusuke Nakata
Graduate School of Science and Engineering
Chiba University
Chiba, Japan
a1019nakata@gmail.com

Sachiyo Arai
Graduate School of Science and Engineering
Chiba University
Chiba, Japan
sachiyo@faculty.chiba-u.jp

ABSTRACT

Reinforcement learning has been applied in numerous decision-making problems; however, it requires a careful specification of a reward function that represents the objective of the problem. There are many problems whose objectives are difficult to represent as a function, and thus, it is easier to give experts' demonstrations for such problems. Inverse reinforcement learning can be applied to such problems because it helps to estimates the reward function from the expert's demonstrations. Most of the existing inverse reinforcement learning methods assume that an expert gives demonstrations in a fixed environment, though the expert can provide demonstrations for a specific objective in multiple environments. For instance, it is difficult to formulate a suitable reward function to represent car driving and the driver can give demonstrations under multiple situations. In such cases, it is normal to use demonstrations in multiple environments to estimate the expert's reward. We formulated and proposed an algorithm for this problem based on Bayesian inverse reinforcement learning. Experimental results show that the proposed method quantitatively outperforms the existing methods.

KEYWORDS

Inverse Reinforcement Learning; Reinforcement Learning; Bayesian Inference

1 INTRODUCTION

Reinforcement learning has been widely employed in decision-making problems such as robotics and video games [6, 8], but requires a careful specification of a reward function that can appropriately represent the objective of a problem. There are problems whose objectives are difficult to be represented as a function; thus it is easier to give expert's demonstrations in such cases. Inverse reinforcement learning (IRL) [1, 9, 12, 14] is a powerful framework for handling such problems because it can be used to estimate a reward function from the expert's demonstrations. IRL estimates a reward that makes the expert's policy optimal in such a way that the expert's policy can be achieved by learning the optimal policy against an estimated reward.

One of the strong points in using IRL is that the estimated reward function can be transferred to other environments whose dynamics are different from the ones demonstrated by the expert [4]. We can achieve an expert's policy in an environment with any dynamics by learning the optimal policy when the IRL discovers the expert's reward. An adaptation to an unseen environment is required in cases such as transfer between different stages of a video game [10], and between simulation and the real-world environment [11]. Kitani et al. [5] showed that reward transfer works for real data. They estimated the reward function from trajectories of pedestrians in a parking lot, and successfully predicted the pedestrian's trajectory in other parking lots and sidewalks using the estimated reward. While the transfer of reward is useful, there is no guarantee that IRL will discover an expert's reward. This problem is caused by the existence of an infinite number of rewards which makes the expert's policy optimal [9].

This paper presents how an expert's data can be applied for estimating transferable reward, which is not the case with most existing IRL methods. To estimate a transferable reward function, we propose the use of an expert's demonstrations in multiple environments with different dynamics (e.g., different stages of a video game) while assuming that expert's rewards are the same across the environments. This is motivated by an assumption that utilization of expert's demonstration in multiple environments helps to reduce uncertainty in reward estimation caused by the existence of multiple rewards which makes the expert's policy optimal [9].

We formulate the problem of reward estimation from demonstrations conducted in multiple environments using the Bayesian inverse reinforcement learning (BIRL) framework [12]. Furthermore, we propose a Markov chain Monte Carlo (MCMC) method for the formulated problem and show that the proposed method has the same speed of convergence to the equilibrium distribution with BIRL setting. From the experiment conducted, we observe that that the method outperforms BIRL in the quantitative evaluation with respect to the expected value difference [7].

2 PRELIMINARIES

2.1 Markov Decision Process (MDP)

In this section, we present the definitions used in the paper. A finite-state Markov decision process (MDP) $\mathcal{M} = (E, R)$, is consists of an environment $E = \langle \mathcal{S}, \mathcal{A}, T, \gamma \rangle$, and a reward R, where $\mathcal{S}$ denotes a finite set of states, $\mathcal{A}$ denotes a finite set of actions, $T(s'|s,a)$ represents the probability of transition to $s' \in \mathcal{S}$ when the agent take an action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$, $\gamma \in (0,1]$ denotes the discount factor, and reward function $R : \mathcal{S} \to \mathbb{R}$ specifies the reward received in state $s \in \mathcal{S}$. An agent can decide to take an action a in state s with the probability specified by the policy $\pi : \mathcal{S} \times \mathcal{A} \to [0,1]$.

A state value and an action value under the reward function R, and policy π are defined as,

$$V^{\pi}(s,R) = \mathbb{E}_{\pi,T}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \middle| s_t = s\right] \tag{1}$$

$$Q^{\pi}(s,a,R) = \mathbb{E}_{\pi,T}\left[\sum_{t=0}^{\infty}\gamma^t R(s_t)\Big|s_t = s, a_t = a\right]. \tag{2}$$

The optimal policy maximizes these values, and the state value for the optimal policy π^* must satisfy the following expression:

$$V^*(s,R) = R(s) + \gamma\sum_{a}\pi^*(a|s)\sum_{s'}T\left(s'|s,a\right)V^*\left(s',R\right). \tag{3}$$

Using the optimal state value V^* defined above, the optimal action value Q^* can be expressed as

$$Q^*(s,a,R) = R(s) + \gamma\sum_{s'}T\left(s'|s,a\right)V^*\left(s',R\right) \tag{4}$$

Herein, we define an optimal policy for the probability distribution of reward $P(R)$, because the proposed method can be used to estimate the $P(R)$. $P(R)$ is useful because it can represent the uncertainty in the reward estimation. A loss of policy π under the distribution $P(R)$ can be defined as

$$L^{p}_{policy}\left(P(R),\pi\right) = \mathbb{E}_{P(R)}\left[\left\|\boldsymbol{V}^*(R) - \boldsymbol{V}^{\pi}(R)\right\|_p\right] \tag{5}$$

where $\boldsymbol{V}^{\pi}(R)$ denotes the vectorized state values of policy π under reward R. p represents the arbitrary norm. A policy that minimizes (5) is an optimal policy for MDP, $\mathcal{M} = (E, \mathbb{E}[R])$ [12].

2.2 Bayesian Inverse Reinforcement Learning(BIRL)

BIRL can be used to estimate a posterior distribution $P(R|D)$, where $D = \{(s_i, a_i)\}_{i=1}^{N}$ represents a dataset of pairs of state s and action a [12]. From Bayes' theorem, the posterior distribution can be expressed as

$$P(R|D) = \frac{P(D|R)}{P(D)}P(R). \tag{6}$$

The likelihood $P(D|R)$ denotes models using the optimal action-value $Q^*(s,a,R)$, which can be expressed as Equation (4). Equation (7) defines the likelihood $P(D|R)$.

$$P(D|R) = \frac{1}{Z}\exp\left(\alpha\sum_{(s,a)\in D}Q^*(s,a,R)\right). \tag{7}$$

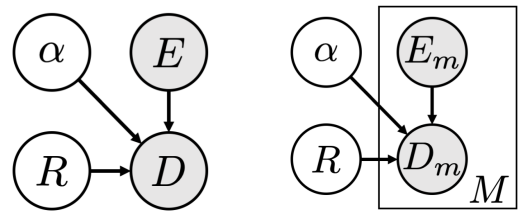
Where Z is a partition function, and α represents a parameter that controls the probability of taking an optimal action. α corresponds to a reciprocal number of temperature for the Boltzmann policy. Using Equations (6) and (7), the posterior distribution $P(R|D)$ can be expressed as

$$P(R|D) = \frac{1}{Z'}\exp\left(\alpha\sum_{(s,a)\in D}Q^*(s,a,R)\right)P(R). \tag{8}$$

$P(R|D)$ is difficult to compute because the partition function Z' is intractable; however, the numerator of this equation can be evaluated by computing optimal action values $Q^*(s,a,R)$. Because

$$P(R|D) \propto \exp\left(\alpha\sum_{(s,a)\in D}Q^*(s,a,R)\right)P(R), \tag{9}$$

we can sample the rewards from posterior distribution $P(R|D)$ using an MCMC method. BIRL introduced an MCMC algorithm, called PolicyWalk, which helps to omit the unnecessary optimal action-value computation. This omission makes PolicyWalk more efficient than the standard MCMC algorithms such as GridWalk [13].



(a) Graphical model for BIRL (b) Graphical model for BIRL-MD

Figure 1: Graphical models for IRL Problems

3 BAYESIAN INVERSE REINFORCEMENT LEARNING FOR MULTIPLE DYNAMICS (BIRL-MD)

The Bayesian inverse reinforcement learning for multiple dynamics (BIRL-MD) defines the problem of estimating the distribution of reward from the demonstrations of an expert under multiple dynamics. A comparison between the of graphical models for BIRL and BIRL-MD is presented in Figure 1.

As discussed in the previous section, BIRL involves estimating the posterior distribution of reward $P(R|D) = P(R|D,E)$ given an environment E and a dataset D, which is generated by an expert in a fixed environment E. However, the proposed BIRL-MD involves estimating the posterior distribution of reward $P(R|\{(D_m,E_m)\}_{m=1}^{M})$ given the environments with different dynamics $E_m = \langle\mathcal{S},\mathcal{A},T_m,\gamma\rangle$ and a set of datasets $\{D_m\}_{m=1}^{M}$ which is generated by an expert in each environment. As shown in Figure 1, we assume that the reward is independent of the environment.

The posterior distribution of the reward given the dataset of the expert $\{(E_m,D_m)\}_{m=1}^{M}$ can be expressed as

$$P(R|\{(D_m,E_m)\}_{m=1}^{M}) = \frac{\Pi_{m=1}^{M}P(D_m|R,E_m)}{\Pi_{m=1}^{M}P(D_m|E_m)}P(R). \tag{10}$$

Herein, we extend the optimal action value $Q^*(s,a,R)$ shown in Equation (7) to the optimal action-value $Q^*(s,a,R,E)$ in order to handle its dependence on the environment. Then, the posterior distribution, $P(R|\{(D_m,E_m)\}_{m=1}^{M})$, can be expressed as

$$P(R|\{(D_m,E_m)\}_{m=1}^{M}) = \frac{1}{Z'}\exp\left(\alpha\sum_{m=1}^{M}\sum_{(s,a)\in D_m}Q^*(s,a,R,E_m)\right)P(R). \tag{11}$$

4 PROPOSED APPROACH

Herein, we propose a method for solving the BIRL-MD problem. We propose an MCMC method that enables sampling rewards from the posterior distribution in a BIRL-MD setting. The numerator

Algorithm 1 PolicyWalk for Multiple Dynamics

INPUT: Environments $\{E_m\}_{m=1}^M$, Demonstrations $\{D_m\}_{m=1}^M$, Prior $P(R)$, Step Size δ
OUTPUT: Sampled Rewards $\{R_i\}_{i=1}^N$

Pick a random vector $R_0 \in \mathbb{R}^{|S|}/\delta$
$\{\pi_m\}_{m=1}^M \leftarrow \{\text{PolicyIteration}(E_m, R_0)\}_{m=1}^M$
for $i = 1$ **do** N
  Pick a reward vector $\tilde{R}$ uniformly at random from the neighbours of $R_{i-1} \in \mathbb{R}^{|S|}/\delta$
  Compute $Q^\pi(s, a, \tilde{R}, E) \quad \forall\{s, a, (E_m, \pi_m)\} \in \mathcal{S} \times \mathcal{A} \times \{(E_m, \pi_m)\}_{m=1}^M$
  if $\exists\{s, a, (E, \pi)\} \in \mathcal{S} \times \mathcal{A} \times \{(E_m, \pi_m)\}_{m=1}^M, Q^\pi(s, \pi(s), \tilde{R}, E) < Q^\pi(s, a, \tilde{R}, E)$ **then** ▷ If any policy is not optimal
    $\{\tilde{\pi}_m\}_{m=1}^M \leftarrow \left\{\text{PolicyIteration}(E_m, \tilde{R})\right\}_{m=1}^M$
    $R_i \leftarrow \tilde{R}$ and $\{\pi_m\}_{m=1}^M \leftarrow \{\tilde{\pi}_m\}_{m=1}^M$ with probability $\min\left\{1, \frac{P\left(\tilde{R}, \{(D_m, E_m)\}_{m=1}^M\right)}{P\left(R_{i-1}, \{(D_m, E_m)\}_{m=1}^M\right)}\right\}$
  else
    $R_i \leftarrow \tilde{R}$ with probability $\min\left\{1, \frac{P\left(\tilde{R}, \{(D_m, E_m)\}_{m=1}^M\right)}{P\left(R_{i-1}, \{(D_m, E_m)\}_{m=1}^M\right)}\right\}$
  end if
end for

of Equation (11) can be computed by learning the optimal action-value $Q^*(s, a, R, E)$ for each environment E_m. This makes it possible to sample from the posterior distribution $P(R|\{(D_m, E_m)\}_{m=1}^M)$ using an MCMC algorithm.

We propose an MCMC algorithm, PolicyWalk for multiple dynamics (PolicyWalk-MD), which is an extension of the PolicyWalk proposed in [12] in order to handle multiple environments. An entire procedure of the PolicyWalk-MD is detailed in Algorithm 1.

PolicyWalk [12] can be used to sample rewards from $P(R|D, E)$ within an error limit of ϵ, in $O(|\mathcal{S}|^2 \log \frac{1}{\epsilon})$ steps. We show that PolicyWalk-MD samples reward as fast as PolicyWalk regardless the number of environments used for reward estimation.

Lemma 4.1. *Let F be a positive real value function defined on $\{x \in \mathbb{R}^n | -d \leq x_i \leq d\}$, where d denotes an arbitrary positive real number. If $f(\cdot) = \log F(\cdot)$ satisfies*

$$|f(x) - f(y)| \leq \alpha_f \|x - y\|_\infty \tag{12}$$

$$f(\lambda x + (1-\lambda)y) \geq \lambda f(x) + (1-\lambda) f(y) - \beta_f \tag{13}$$

for all $\lambda \in [0, 1]$ and α_f, β_f, the Markov chain induced by GridWalk, PolicyWalk, (and hence PolicyWalk-MD) on F rapidly mixes to within ϵ of F in $O(n^2 d^2 \alpha_f^2 e^{2\beta_f} \log \frac{1}{\epsilon})$ [3, 12].

Theorem 4.2. *Let $E = \langle \mathcal{S}, \mathcal{A}, T, \gamma \rangle$ be an Environment, and a prior distribution of reward $P(R)$ is uniform, i.e., $\mathcal{U}(-R_{\max}, R_{\max})$, and posterior $P(R|\{(D_m, E_m)\}_{m=1}^M)$ is defined as Equation (11). Suppose $R_{\max} = O(\frac{1}{M|D|})$, PolicyWalk-MD's sampling can be rapidly mixed to within an error limit ϵ of $P(R|\{(D_m, E_m)\}_{m=1}^M)$ in $O(|\mathcal{S}|^2 \log \frac{1}{\epsilon})$ steps.*

Proof.

$$f(R) = \alpha \sum_{m=1}^{M} \sum_{(s,a) \in D_m} Q^*(s, a, R, E_m) \tag{14}$$

$$f_\pi(R) = \alpha \sum_{m=1}^{M} \sum_{(s,a) \in D_m} Q^\pi(s, a, R, E_m) \tag{15}$$

where f_π is linear for vector R, and $f(R) \geq f_\pi(R)$ for arbitrary reward R. For the action value $Q(s, a, R, E_m)$, both

$$\max_{s,a} Q^*(s, a, R, E) \leq \sum_{t=0}^{\infty} \gamma^t R_{\max} = \frac{R_{\max}}{1-\gamma} \tag{16}$$

and $\min_{s,a} Q^*(s, a, R, E) \geq -\frac{R_{\max}}{1-\gamma}$ hold. Using these Equations, we have

$$f_\pi(R) \geq -\frac{\alpha M|D|R_{\max}}{1-\gamma}, \tag{17}$$

$$\frac{\alpha M|D|R_{\max}}{1-\gamma} \geq f(R), \tag{18}$$

$$-\frac{\alpha M|D|R_{\max}}{1-\gamma} \geq f(R) - \frac{2\alpha M|D|R_{\max}}{1-\gamma}. \tag{19}$$

By inseting Equation (17) into Equation (19), we get

$$f_\pi(R) \geq f(R) - \frac{2\alpha M|D|R_{\max}}{1-\gamma}. \tag{20}$$

Therefore,

$$f(\lambda R_1 + (1-\lambda)R_2) \geq f_\pi(\lambda R_1 + (1-\lambda)R_2) \tag{21}$$

$$= \lambda f_\pi(R_1) + (1-\lambda) f_\pi(R_2) \tag{22}$$

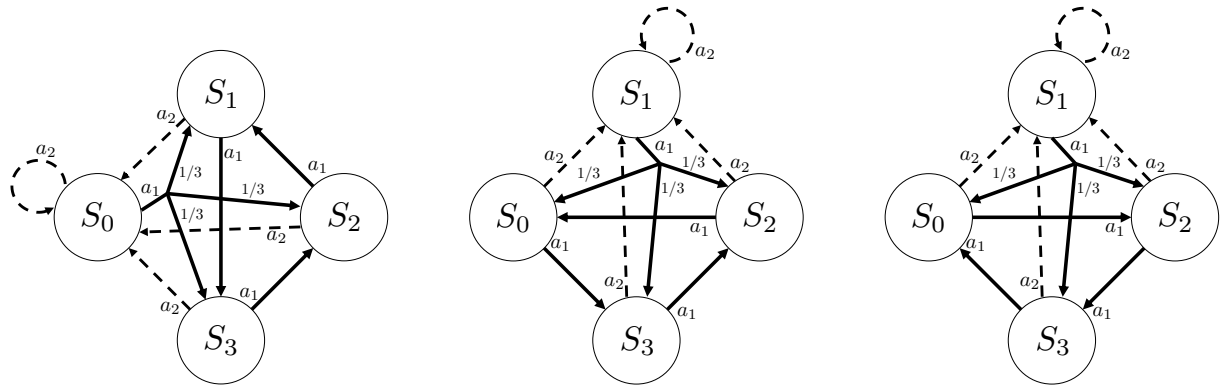
$$\geq \lambda f(R_1) + (1-\lambda) f(R_2) - \frac{2\alpha M|D|R_{\max}}{1-\gamma}, \tag{23}$$

and the following conditions hold for variables in the lemma:

$$\alpha_f = \frac{|f(R_1) - f(R_2)|}{\|R_1 - R_2\|_\infty} \leq \frac{2\alpha M|D|R_{\max}}{(1-\gamma)O\left(\frac{1}{M|D|}\right)} = O(M|D|), \tag{24}$$

$$\beta_f = \frac{2\alpha M|D|R_{\max}}{1-\gamma} = 2\alpha M|D| \frac{O\left(\frac{1}{M|D|}\right)}{1-\gamma} = O(1). \tag{25}$$

Hence, the Markov chain induced by the PolicyWalk-MD on $P(R|\{(D_m, E_m)\}_{m=1}^M)$ mixes rapidly within an error limit ϵ of



(a) Environment A (b) Environment B (c) Environment C

Figure 2: Environments and expert's policy for the experiments. Bold lines represent the optimal action a_1, and broken lines represent other action a_2. Numbers beside the lines indicate transition probability.

$P(R|\{(D_m, E_m)\}_{m=1}^{M})$ in a number of steps equal to

$$O\left(|\mathcal{S}|^2 \frac{1}{(M|D|)^2} (M|D|)^2 \log \frac{1}{\epsilon}\right) = O\left(|\mathcal{S}|^2 \log \frac{1}{\epsilon}\right). \quad (26)$$

□

Notably, $R_{\max} = O(\frac{1}{M|D|})$ is not really a restriction because the rewards can be rescaled by a constant factor after computing the mean without changing the optimal policy. The step size δ in the Algorithm 1 is introduced to split the real $|\mathcal{S}|$-space $\mathbb{R}^{|\mathcal{S}|}$ as GridWalk and PolicyWalk do [12, 13].

5 EVALUATION

5.1 Experimental Setting

In this section, we compare BIRL and BIRL-MD on a simple IRL problem whose reward estimation is subject to uncertainties. Figure 2 presents the environments for the experiments conducted. We set an expert's true reward as $R(s_0) = R(s_1) = 0, R(s_2) = R(s_3) = 0.7$. The bold line in the Figure 2 represents an expert's action which maximizes the expected discounted return. Reward estimation is subject to uncertainty because the expert's policies are described using multiple reward functions. For example, in the environment A, any reward that sets a high reward value to any combination of s_1, s_3, s_2 describes the expert's policy as an expert in this environment rounds s_1, s_3, and s_2. The environment B and C have difference in the transition probabilities. An expert in the environment B rounds states in order of s_0, s_3, and s_2, while an expert in the environment C rounds states in order of s_0, s_2, and s_3.

We use the expected value difference（EVD）[7] for evaluating the similarities between the estimated rewards and the expert's reward. EVD is a measure of how optimal the leaned policy for the estimated reward is under the expert's true reward. To compute EVD, we compare the optimal policy under each estimated reward and expert's policy with the expected discounted reward of the expert's reward.

We evaluate the EVD using 100 test environments with different transition probabilities, which the stochastic transition probabilities are randomly generated using uniform distribution for each test environments. A number of states and actions of test environments are the same as those for the training environments presented in the Figure 2. For EVD evaluation, we created a dataset of state action pairs of the expert is generated from a Boltzmann policy with $\gamma = 0.95$ and the temperature $\kappa = 1/3$. The dataset for each environment comprises 30 trajectories with 10 steps, resulting in 300 state-action pairs in the dataset.

It is necessary to set the parameter α and a prior distribution to be a le to estimate the reward using BIRL and BIRL-MD. In the experiments, we set the value of α with respect to the temperature of expert's policy $\alpha = 1/\kappa = 3$, because α corresponds to the reciprocal of the temperature κ. The prior distribution $P(R)$ was set to the uniform distribution $\mathcal{U}(-1, 1)$. The MCMC steps, burn-in, and the step size δ, are 2000, 200, and 0.01 respectively. We used the mean of the sampled rewards as the estimated reward for both BIRL[12] and BIRL-MD.

5.2 Results

The mean, standard deviation and maximum value of the EVD of the estimated reward for BIRL and BIRL-MD are listed in a Table 1.

Method	BIRL	BIRL-MD(Ours)	
Number of Environments	1	2	3
EVD(Mean±Std)	1.15 ± 0.89	0.17 ± 0.24	**0.02 ± 0.06**
EVD(Max)	3.93	1.18	**0.36**

Table 1: Evaluation of BIRL and BIRL-MD using EVD. BIRL uses only one environment, while BIRL-MD uses multiple environments for estimating the reward. The bold value represents the lowest (best) value of EVD.

The top row in the Table 1 shows the method, while the second row is the number of environments M used for estimation of posterior distribution $P(R|\{(D_m, E_m)\}_{m=1}^{M})$. The third row shows the mean and standard deviation, whereas the fourth row represents the maximum value of the EVD. The EVDs are computed using the test environments and estimated rewards. The reward is estimated using a set of datasets $\{(D_m, E_m)\}_{m=1}^{M}$, which consists of dataset from M-combination of environments shown in Figure 2. For example, the values in the second column are computed using the EVDs for every three rewards, and the datasets obtained from each one of the environments A, B, and C are used exactly once to estimate each reward. For the values of the third column which used two environments, we estimated three rewards and each reward is estimated using the datasets from a combination of three environments in the Figure 2 by considering two environments without repetition, {A, B}, {A, C}, and {B, C}.

As the number of environments for reward estimation increases, the mean, standard deviation and the maximum value of EVD decreases. Since the lower EVD value indicates better performance, it can be said that the BIRL-MD outperforms BIRL in terms of the similarity between estimated reward and the expert's true reward.

6 RELATED WORK

In this study, we focused on how to obtain a transferable reward. Similarly, adversarial IRL (AIRL) [4] can be used to learn transferable reward. They proposed estimation of state-only reward $R(s)$ rather than $R(s, a)$, and $R(s, a, s')$. However, AIRL estimates the reward using demonstrations in a fixed environment, and it cannot estimate the reward using demonstrations from multiple environments. Repeated IRL [2] uses demonstrations from multiple environments, and the motivation for its use is closely related to BIRL-MD. Repeated IRL can be used to estimate a common reward that is shared across multiple tasks, given task-specific rewards, environments, and demonstrations. We defined the problem of reward estimation resulting from demonstrations in multiple environments in a Bayesian manner. It enables us to combine prior knowledge and evidence from the expert to derive a probability distribution over rewards.

7 CONCLUSIONS

In this study, we formulated the problem of Bayesian inverse reinforcement for multiple dynamics, which estimates the expert's reward from demonstrations of an expert under multiple dynamics, and proposed the MCMC method called PolicyWalk for multiple dynamics. We compared the proposed method with BIRL [12] using EVD [7]. Experimental results showed that the proposed method outperforms the BIRL[12] in terms of the similarity between the estimated rewards and the expert's true reward. Although the proposed formulation can be applied applicable to environments with continuous state and action space, the applicability of the method with the MCMC algorithm is limited to small environments with a discrete state action space owing to the huge computational cost involved. It is necessary to develop an algorithm that can solve BIRL-MD problem in environments with continuous and large state action space, which would be the focus of our future study.

REFERENCES

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

[2] Kareem Amin, Nan Jiang, and Satinder Singh. 2017. Repeated Inverse Reinforcement Learning. In *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 1815–1824. http://papers.nips.cc/paper/6778-repeated-inverse-reinforcement-learning.pdf

[3] David Applegate and Ravi Kannan. 1991. Sampling and integration of near log-concave functions. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*. ACM, 156–163.

[4] Justin Fu, Katie Luo, and Sergey Levine. 2017. Learning Robust Rewards with Adversarial Inverse Reinforcement Learning. *arXiv preprint arXiv:1710.11248* (2017).

[5] Kris M Kitani, Brian D Ziebart, James Andrew Bagnell, and Martial Hebert. 2012. Activity forecasting. In *European Conference on Computer Vision*. Springer, 201–214.

[6] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. 2016. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research* 17, 1 (2016), 1334–1373.

[7] Sergey Levine, Zoran Popovic, and Vladlen Koltun. 2011. Nonlinear inverse reinforcement learning with gaussian processes. In *Advances in Neural Information Processing Systems*. 19–27.

[8] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529.

[9] Andrew Y Ng, Stuart J Russell, et al. 2000. Algorithms for inverse reinforcement learning.. In *Icml*. 663–670.

[10] Alex Nichol, Vicki Pfau, Christopher Hesse, Oleg Klimov, and John Schulman. 2018. Gotta Learn Fast: A New Benchmark for Generalization in RL. *arXiv preprint arXiv:1804.03720* (2018).

[11] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. 2017. Robust adversarial reinforcement learning. *arXiv preprint arXiv:1703.02702* (2017).

[12] Deepak Ramachandran and Eyal Amir. 2007. Bayesian inverse reinforcement learning. *IJCAI International Joint Conference on Artificial Intelligence* (2007), 2586–2591.

[13] Santosh Vempala. 2005. Geometric random walks: a survey. *Combinatorial and computational geometry* 52, 573-612 (2005), 2.

[14] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. 2008. Maximum Entropy Inverse Reinforcement Learning.. In *AAAI*, Vol. 8. Chicago, IL, USA, 1433–1438.