Inverse Optimal Control from Demonstrations with Mixed Qualities

Kyungjae Lee, Yunho Choi, and Songhwai Oh

***Abstract*— This paper proposes an inverse optimal control (IOC) framework which incorporates demonstrations with mixed qualities. The proposed method utilizes the benefits of sub-optimal demonstrations which can provide information about *what not to do* and supplies training data near states unvisited by optimal demonstrations. The main idea of the proposed method is to find the value function which satisfies the optimality condition over optimal demonstrations and violates it over sub-optimal demonstrations. We conduct experiments on three environments and empirically show that the proposed method outperforms the original IOC algorithm, which uses only optimal demonstrations.**

I. Introduction

Inverse optimal control (IOC) has enabled for a robot to perform complex tasks, such as inverted helicopter flight [1], autonomous driving [2], [3], and manipulation [4]. An IOC framework learns both reward and policy function from expert's demonstrations. In general, IOC aims to recover a reward function which can generate expert's demonstrations. A basic assumption of IOC is that the expert acts to maximize the expected cumulative underlying reward. From this assumption, the main idea of IOC is to find a reward function under which expert's demonstrations become an optimal solution. In this regards, existing IOC algorithms require solving optimal control problems under the estimated reward as a subroutine, in order to verify if the expert's demonstrations are optimal.

Since most existing algorithms utilize only optimal demonstrations which are often located near the high reward regions, the resulting reward function has no capability to estimate the low reward regions correctly. This issue is highly investigated in several studies [5]–[7]. In [5], the authors argued that insufficient demonstrations near the low reward region will cause a catastrophic failure. Although [5] handles this issue via continuously interacting with an expert who knows the optimal control, it is not viable to contact the expert repeatedly in practice. Instead of constant communication with the expert, [6] and [7] utilize failed and sub-optimal demonstrations, respectively. In [6], the method of incorporating optimal demonstrations and failed demonstrations is proposed by adding constraints that make the resulting behavior maximally different from the failed demonstrations. In [7], Lee et al. incorporate both optimal and sub-optimal demonstrations where the algorithm assigns a high reward near the optimal demonstrations and a low reward near the sub-optimal demonstrations. The sub-optimal demonstrations give an information about *what not to do* near low reward regions. While [6] and [7] have successfully utilized sub-optimal demonstrations for an IOC problem, [6] and [7] require to solve the reinforcement learning (RL) problem as a subroutine to check whether expert's demonstrations become an optimal solution or not. In this regards, [6] and [7] suffer from a heavy computational load and require a significantly large number of sampled demonstrations to solve RL problems.

To handle both sample inefficiency and the lack of demonstrations near the low reward regions, we employ the necessary and sufficient condition of Hamilton-Jacobi-Bellman (HJB) equation. The HJB equation is a well known partial differential equation where the solution is the maximum cumulative reward, also knwon as the optimal value. By assuming control affine dynamics and a control quadratic reward function, a closed-loop optimal policy can be obtained from HJB, i.e., $u = R^{-1}G(x)^T\nabla_x V(x)$. We observe that, by using this optimal policy, the gradient information of the value function can be extracted from expert's actions. The main idea of the proposed method is to find the value function which satisfies the optimal policy for expert's optimal demonstrations and violates its optimality for sub-optimal demonstrations. In this regards, a value function is directly learned from optimal and sub-optimal control data and the resulting policy can be obtained from HJB.

In this paper, we propose a novel inverse optimal control algorithm utilizing both optimal and sub-optimal demonstrations in continuous time, state, and control spaces. The proposed algorithm finds the value function that makes optimal demonstrations to satisfy the optimal policy of HJB and sub-optimal demonstrations to violate the optimality condition. As our method directly estimates the underlying value function, the controller can be directly obtained from the resulting value function without computation of an optimal control problem. For the problem without knowledge of system dynamics, we learn the control affine dynamics using a neural network. Furthermore, for high dimensional state space such as range sensor or image, we employ an encoder from the state space to the low dimensional latent space. Incorporating both optimal and sub-optimal demonstrations improves the generalization performance of the resulting dynamic model since sub-optimal demonstrations cover more various situations than optimal demonstrations. We empirically show that the proposed method outperforms other methods which utilize only optimal demonstrations in terms of the average cumulative reward and achieves near optimal

K. Lee, Y. Choi, and S. Oh are with the Department of Electrical and Computer Engineering and ASRI, Seoul National University, Seoul 08826, Korea (e-mail: {kyungjae.lee, yunho.choi}@cpslab.snu.ac.kr, songhwai@snu.ac.kr). This work was supported by the ICT R&D program of MSIT/IITP (No. 2019-0-01309, Development of AI Technology for Guidance of a Mobile Robot to its Goal with Uncertain Maps in Indoor/Outdoor Environments).

performance with a smaller number of demonstrations than other IOC algorithms.

II. Related Work

Inverse optimal control problems have been widely studied in robotics and machine learning [6]–[18]. Existing IOC algorithms can be categorized into four classes based on two criteria. The first criterion is the necessity of solving an optimal control problem as a subroutine. A majority of IOC methods compute the optimal control in order to update estimated reward function, while [16]–[18] find a reward function without solving an optimal control problem. Those methods often utilize extra optimality criterion such as the HJB equation, similar to the proposed method. The second criterion is the capability of handling both optimal and sub-optimal demonstrations. A number of existing IOC algorithms deal with only optimal demonstrations and a handful number of methods including ours incorporate both optimal and sub-optimal demonstrations.

Most IOC algorithms are formulated based on MDPs and these methods require to solve an MDP problem under the estimated reward function in order to compute the gradient of the reward function. Unlike other MDP-based IOC algorithms, in [12], Ho and Ermon proposed the generative adversarial imitation learning (GAIL) method where the policy function is updated to maximizes the reward function and the reward function is updated to assign high values to expert's demonstrations and low values to trained policy's demonstrations. GAIL achieves sample efficiency by avoiding the need to solve RL as a subroutine and alternatively updating policy and reward functions. However, since GAIL utilizes a policy gradient method to optimize a policy function, it still requires a large number of samples to converge stably to a stationary point.

On contrary, several IOC algorithms are formulated based on the Euler-Lagrange (EL) equation and Hamilton-Jacobi-Bellman (HJB) equation in optimal control [16]–[19]. These algorithms do not require to solve optimal control problems since they utilize inverse optimality conditions induced from the EL and HJB equations. These algorithms often aim to find the reward function which satisfies some inverse optimality condition on the given demonstrations. In [17], an inverse optimal control algorithm using polynomial optimization is proposed where the sufficient condition of the HJB equation is used to formulate an IOC problem and the reward function is modeled as a linear combination of polynomial functions. In [18], Puydupin-Jamin et al. proposed an inverse optimal control algorithm minimizing the residual function defined based on the Karush-Kuhn-Tucker (KKT) necessary conditions where it is equivalent to satisfying the optimality condition of the residual function being zero.

There are a few methods which can learn from both optimal and sub-optimal demonstrations. A more detailed study is necessary for utilizing sub-optimal demonstrations. [15] proposed semi-supervised apprenticeship learning (SSAL) which utilizes both optimal demonstrations and unlabeled demonstrations which may not be from an expert. SSAL simultaneously finds a reward function and classifies unlabeled demonstrations using the maximum margin based method. In [13], maximum entropy semi-supervised inverse reinforcement learning (MESSIRL) which utilizes both labeled and unlabeled demonstrations based on MaxEnt IRL is proposed, where the inexpert examples in unlabeled demonstrations are filtered by the similarity-based penalty. Similarly, [14] proposed robust Bayesian inverse reinforcement learning (RBIRL) which can automatically identify and eliminate noisy demonstrations. SSAL, MESSIRL, and RBIRL are more focused on categorizing unclassified demonstrations. Since the method incorporating sub-optimal demonstrations [13]–[15] are formulated based on [8], [9], these algorithms also need to solve an MDP or RL problem as a subroutine in each iteration.

III. Background

In this paper, we consider a continuous-time control affine system defined as $\dot{x} = F(x) + G(x)u$, where $x \in \mathbb{R}^{d_X}$ and $u \in \mathbb{R}^{d_U}$ are state and control variables, $F(x) \in \mathbb{R}^{d_X \times d_X}$ and $G(x) \in \mathbb{R}^{d_X \times d_U}$ are matrix functions of state. The reward function is assumed as a control quadratic $l(x,u) = q(x) - \frac{1}{2}u^T R u$, where $q(x)$ is a state dependent reward function and $R \in \mathbb{R}^{d_U \times d_U}$ is a positive semi-definite matrix. Based on the control affine dynamic system and control quadratic reward function, we first explain an optimal control problem and the HJB equation which is the necessary and sufficient condition for a solution to be optimal. Then, the inverse optimal control problem is explained.

A. Optimal Control and HJB Equation

The optimal control problem is to find the optimal policy to maximize the infinite horizon discounted cumulative reward defined as follows:

$$J(x_0, u(\cdot)) = \int_0^\infty e^{-t/\tau} \left[q(x(t)) - \frac{1}{2} u(t)^T R u(t) \right] dt, \qquad (1)$$

where x_0 is an initial state, $\tau > 0$ is a constant discount factor, $u(\cdot)$ is a control functional depending on time t. Under this performance criterion, the optimal control problem is defined as follows:

$$\begin{aligned} \underset{u(\cdot)}{\text{maximize}} \quad & J(x_0, u(\cdot)) \\ \text{subject to} \quad & \dot{x}(t) = F(x(t)) + G(x(t))u(t),\ x(0) = x_0. \end{aligned} \qquad (2)$$

The maximum performance criterion is referred to as the value function defined as $V(x_0) = \max_{u(\cdot)} J(x_0, u(\cdot))$. The value function indicates the performance of the optimal control functional. The HJB equation is a partial differential equation of the value function defined as

$$\tau^{-1} V(x(t)) = \max_{u(t)} \mathcal{H}(x(t), u(t), V(x(t))), \qquad (3)$$

where $\mathcal{H}$ is the Hamiltonian defined as

$$\begin{aligned} &\mathcal{H}(x, u, V(x)) = \\ &q(x) - \frac{1}{2} u^T R u + \nabla_x V(x)^T \left(F(x) + G(x)u \right), \end{aligned} \qquad (4)$$

where ∇_x indicates the gradient of the value function with respect to the state x. Since the HJB equation is the necessary and sufficient condition, the optimal policy must satisfy the HJB equation. In case of the control quadratic reward and control affine dynamics, the optimal policy is explicitly obtained as the following closed-loop control,

$$\phi(x) \triangleq u^* = R^{-1}G(x)^T\nabla_x V(x) \tag{5}$$

by finding the maximum of Hamiltonian $\mathcal{H}$ where $*$ indicates the optimal solution. By substituting the resulting policy into (4), we obtain the relation between state dependent reward function and the value function as follows:

$$\begin{aligned} q(x) =& \frac{1}{\tau}V(x) - \nabla_x V(x)^T F(x) \\ & - \frac{1}{2}\nabla_x V(x)^T G(x) R^{-1} G(x)^T \nabla_x V(x). \end{aligned} \tag{6}$$

In this paper, we assume that the expert obeys the closed-loop optimal policy (5) maximizing the underlying control quadratic reward function. Hence, optimal demonstrations are generated from the optimal policy and sub-optimal demonstrations are obtained from a non-optimal policy.

B. Inverse Optimal Control

In this section, we introduce an inverse optimal control problem whose goal is to estimate a reward function making expert's demonstrations an optimal solution. Several existing continuous IOC algorithms including the proposed method assume that expert obeys the closed-loop optimal policy in (5) [16]–[18].

Particularly, we follow [16] where the gradient of the value function is directly estimated from demonstrations. Expert's demonstrations consist of a sequence of state and corresponding optimal action pairs, i.e $\mathcal{D} = \{\zeta_i\}_{i=0}^N$ and $\zeta_i = \{(x_{i,t}, u_{i,t})\}_{t=0}^T$, where N is the number of demonstrations and T is the length of a sequence of each demonstration. In [16], the value function is modeled as a linear combination of squared exponential (SE) kernel functions which are defined as

$$\hat{V}(\cdot) = \sum_{j=0}^{M} k_{SE}(\cdot, y_j)\alpha_j, \tag{7}$$

where α_j is the jth coefficient of the jth SE kernel, y_j is a predefined center point of the kernel, M is the number of total predefined center points, and

$$k_{SE}(x, x') = \beta \exp\left(-||x - x'||2/(2\sigma^2)\right). \tag{8}$$

Here, $\beta > 0$ and σ are hyperparameter. [16] finds the coefficients of SE kernels which satisfy the optimal policy (5) given demonstrations $\mathcal{D}$. The optimization problem in [16] is defined as follows:

$$\min_{\alpha} \mathcal{L}(\mathcal{D}, \hat{V}) \triangleq \sum_{i=0}^{N}\sum_{t=0}^{T} ||u_{i,t} - R^{-1}G(x_{i,t})^T\nabla_x\hat{V}(x_{i,t})||_2^2, \tag{9}$$

where R is often assumed as identity matrix. The gradient of $\hat{V}$ is computed as

$$\nabla_x\hat{V}(x) = \sum_{j=0}^{M} \nabla_x k_{SE}(x, y_j)\alpha_j, \tag{10}$$

where $\nabla_x k_{SE}$ indicates the gradient of the SE kernel with respect to the first input. [16] converts an IOC problem into a kernel regression problem with gradient information and solves it using the least square method. After estimating the value function, the state dependent reward function is obtained from (6). This algorithm has benefits in that the learned value function directly induces the controller without solving the optimal control problem under the recovered reward function.

The major drawback of [16] is that the SE kernel function has an improper property to represent the gradient of the value function. The gradient of the SE kernel function is as follows:

$$\nabla_x k_{SE}(x, y_i) = \frac{\beta}{\sigma^2}(x - y_i)\exp\left(-||x - y_i||2/(2\sigma^2)\right). \tag{11}$$

Since the gradient of the SE kernel function is zero at $x = y_i$ and also goes to zero as the distance between x and y_i goes to infinity, it leads to poor approximation. To handle this issue, we model the value function using neural networks with a smooth activation function.

IV. Inverse Optimal Control from Demonstrations with Mixed Qualities

In this section, we propose a novel inverse optimal control from demonstrations with mixed qualities (IOCfDMQ) which has a capability to incorporate both optimal and sub-optimal demonstrations. In our problem, it is assumed that the positive demonstrations are a set of sequences that consists of state and optimal control pairs, i.e., $\mathcal{D}^+ = \mathcal{D}$, and the sub-optimal demonstrations consists of a sequence of state and sub-optimal control pairs, i.e., $\mathcal{D}^- = \{(x_{i,t}, u_{i,t})_{t=0}^{T_i}\}_{i=0}^{N^-}$, where N^- is the number of sub-optimal demonstrations, $+$ and $-$ indicate optimal and sub-optimal data, respectively.

The main idea of the proposed method is to extract meaningful information about *what not to do* from sub-optimal demonstrations in continuous state and control spaces. We first formulate our problem by adding a new constraint to (9) and reformulate the problem using relaxation technique. We further explain the benefits of using negative demonstrations and also propose a model free version of IOCfDMQ by learning dynamics from demonstrations.

A. Problem Formulation

We consider the problem of finding the value function to generates given positive demonstrations and not to create negative demonstrations. The proposed IOC framework can be formulated as follows:

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & \mathcal{L}^+(\mathcal{D}^+, \hat{V}_\theta) + \lambda B(\hat{V}_\theta) \\ \text{subject to} \quad & \forall\ x_{i,t}, u_{i,t} \in \mathcal{D}^- \\ & ||u_{i,t} - R^{-1}G(x_{i,t})^T\nabla_x\hat{V}_\theta(x_{i,t})||_2^2 \geq \epsilon, \end{aligned} \tag{12}$$

where θ is parameters of a value estimator $\hat{V}_\theta$, $B(\hat{V}_\theta)$ is a regularization function, $\lambda > 0$ is a regularization coefficient, and $\mathcal{L}^+(\mathcal{D}^+, \hat{V}_\theta) = \mathcal{L}(\mathcal{D}^+, \hat{V}_\theta)$, the same as in (9). However, we newly add the constraint that requires the Euclidean distance between the control from sub-optimal demonstrations

and the estimated policy to be greater than the threshold ϵ. This constraint comes from the optimality condition of the HJB equation.

The Hamiltonian function can be rewritten using the optimal policy (5) as follows:

$$\begin{aligned}\mathcal{H}(x,u,V(x)) &= q(x) - \frac{1}{2}(u-\phi(x))^T R(u-\phi(x)) \\ &+ \nabla V(x)^T F(x) + \frac{1}{2}\phi(x)^T R\phi(x).\end{aligned} \tag{13}$$

The maximum of the $\mathcal{H}$ is achieved when $(u-\phi(x))^T R(u-\phi(x))$ is zero for the fixed state x since $\mathcal{H}$ is a quadratic function with respect to the control u. Our goal is to find the value function whose policy satisfies the equality $(u-\phi(x))^T R(u-\phi(x)) = 0$ for $\mathcal{D}^+$ and the inequality $(u-\phi(x))^T R(u-\phi(x)) > 0$ for $\mathcal{D}^-$, respectively. From the positive definiteness of matrix R, it can be shown that $(u-\phi(x))^T R(u-\phi(x)) = 0$ is equivalent to $(u-\phi(x))^T(u-\phi(x)) = 0$ since the following inequalities hold: $\lambda_{\max}||u-\phi(x)||_2^2 \geq (u-\phi(x))^T R(u-\phi(x)) \geq \lambda_{\min}||u-\phi(x)||_2^2$ where $\lambda_{\max}$ and $\lambda_{\min}$ are the maximum and minimum eigenvalues of R, respectively, and are always positive. Conversely, $(u-\phi(x))^T R(u-\phi(x)) > 0$ is also equivalent to $(u-\phi(x))^T(u-\phi(x)) > 0$. From this fact, the objective function and constraint in (12) are induced. Minimizing the objective function let the resulting policy dervied from the learned value function to be close to the optimal control in $\mathcal{D}^+$ and the constraints make the resulting policy differ from the sub-optimal control in $\mathcal{D}^-$.

Since the feasible set of problem (12) is a non-convex set, which is defined by the upper-level set of a quadratic function, this problem is non-convex. To handle this issue, the inequality constraint is relaxed by introducing

$$\begin{aligned}&\mathcal{L}^-(\mathcal{D}^-, \hat{V}_\theta, \epsilon) \triangleq \\ &\frac{1}{2}\sum_{i=0}^{N^-}\sum_{t=0}^{T}\left[\epsilon - ||u_{i,t} - R^{-1}G(x_{i,t})^T\nabla_x \hat{V}_\theta(x_{i,t})||_2^2\right]_+,\end{aligned}$$

where $[x]_+ = \max(x, 0)$ is the hinge loss, ϵ is the margin between the estimated policy and the tth sub-optimal control in the ith demonstration. $\mathcal{L}^-$ becomes zero when the distance between the estimated policy and the sub-optimal control is below the predefined threshold. In other words, if the optimization variables satisfy constraints in the original problem (12), then $\mathcal{L}^-$ will be zero. Consequently, we aim to solve the following relaxed IOC problem,

$$\underset{\theta}{\text{minimize}} \quad \mathcal{L}^+(\mathcal{D}^+, \hat{V}_\theta) + \lambda^-\mathcal{L}^-(\mathcal{D}^-, \hat{V}_\theta, \epsilon) + \lambda B(\hat{V}_\theta), \tag{14}$$

where λ^- is a coefficient.

Note that, since minimizing $\mathcal{L}^+$ is equivalent to align the gradient of the value estimator to the direction of the optimal control, the estimated value function increases near $\mathcal{D}^+$. Conversely, minimizing $\mathcal{L}^-$ ensures that the gradient of the estimated value function does not align in the direction similar to the sub-optimal control. In this regards, the estimated value function does not increase near $\mathcal{D}^-$. We would like to emphasize that the proposed method does not require to solve an optimal control problem as a subroutine since it utilizes the relationship between the optimal policy and the value function from the HJB equation.

To obtain a smooth estimation result, we minimize the output of the value function using the following regularization: $B(\hat{V}_\theta) \triangleq \sum_{x\in\mathcal{D}}||\hat{V}_\theta(x)||_2^2$, where $\mathcal{D}$ is entire training demonstrations. Since we only utilize the gradient information of the value function, the constant of the indefinite integral cannot be determined in our optimization. In this regards, the regularization helps to handle such illposedness and penalizes the scale of the output value to increase.

B. Model-Free Case

While the proposed method is well defined when control affine dynamics are known, in many complex problems, control affine dynamics are often unavailable. In this case, we learn a control affine dynamics using a neural network, which is called a dynamic network. The network output consists of three components: passive dynamics $F_\nu(x)$, control dynamics $G_\nu(x)$ and variance $\Sigma_\nu(x)$ of dynamics, where $\Sigma_\nu(x)$ is a diagonal matrix and ν is the parameter of a dynamic network. A dynamic network is trained using the negative log-likelihood loss funciton which is defined as follows:

$$\mathcal{L}_{\text{dyn}}(\mathcal{D},\nu) \triangleq \frac{1}{2|\mathcal{D}|}\sum_{(x_t,u_t,x_{t+1})\in\mathcal{D}} -\log(p_\nu(x_{t+1}|x_t,u_t)),$$

where $\mathcal{D}$ indicates entire transition pairs including both optimal and sub-optimal demonstrations and $p(x_{t+1}|x_t,u_t) = \mathcal{N}(x_{t+1}; x_t + F_\nu(x_t) + G_\nu(x_t)u_t, \Sigma_\nu(x_t))$.

To summarize, when the dynamic model is unknown, we first train a dynamic network. After training dynamic network, the value network is trained using (14) with the trained dynamic model.

V. Experiments

To validate the effectiveness of sub-optimal demonstrations, we prepare two experiments: a model based problem and a model free problem. In the model based problem, we demonstrate that the proposed method has a capability of recovering the underlying optimal policy. In the model free problem, we first verify that sub-optimal demonstrations help to train a dynamic network in terms of the generalization performance and compare the proposed method with the method using only optimal demonstrations [16] with respect to the average returns. We model the value function using a multi-layered perceptron with two hidden layers with 64 hidden units where a tangent hyperbolic and cosine hyperbolic functions are used for the activation functions for the first and second layer, respectively. In the model free problem, the dynamic network is also modeled as a two layered perceptron with 64 hidden units using a tangent hyperbolic activation. Both the proposed and the compared method use the same network architecture for the entire experiments and the other hyperparameters are determined by a brute force search.

A. Model Based Problem

In this simulation, we compare the proposed method to the method using only optimal demonstrations. We call the

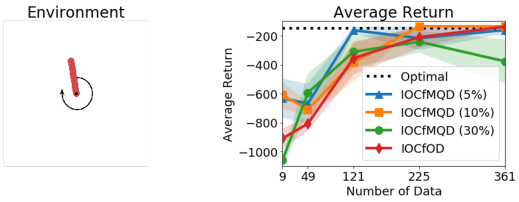


Fig. 1: An illustration of the pendulum environment and average returns of different algorithms.

latter an inverse optimal control from optimal demonstrations (IOCfOD) [16]. We test all algorithms on the pendulum problem as shown in the left figure in Figure 1. The system has two state variables which are the angle from the vertical line and its angular velocity, i.e., $[\theta, \dot{\theta}]$, and has one control variable u which is the torque at the joint. The control affine dynamics is $\ddot{\theta} = -\frac{3g}{2l}\sin(\theta+\pi)+\frac{3}{ml^2}u$ where $g = 10.0$ is the gravity, $l = 1.0$ is the length of the pendulum, and $m = 1.0$ is its mass. The matrix G is $[0, \frac{3}{ml^2}]^\intercal$. The underlying reward function is defined as: $r(\theta, \dot{\theta}, u) = -\left(\frac{\theta}{\pi}\right)^2 - 0.1\dot{\theta}^2 - 0.001u^2$. The optimal demonstrations are generated by controlling the agent with the optimal policy which is obtained by solving the HJB equation. The sub-optimal demonstrations are generated by flipping the optimal control, where the flipped control is clearly sub-optimal. We prepare three different sets of demonstrations by mixing optimal and sub-optimal demonstrations with three different ratios: (95%, 5%), (90%, 10%), and (70%, 30%) and measure the performance while increasing the number of given state action pairs.

The results are shown in Figure 1. The proposed methods with ratio 5% and 10% outperform IOCfOD which uses only optimal demonstrations in terms of the number of demonstrations required to achieve the expert's performance. Especially, the proposed method with 5% sub-optimal demonstrations achieves the expert's performance using the smallest number of data.

Based on this experimental result, we can conclude that incorporating both optimal and sub-optimal demonstrations has a benefit in terms of average returns and it is empirically shown that there exists the best ratio that improves the performance of the resulting controller using sub-optimal demonstrations.

B. Model Free Problem

The simulations are conducted in the MuJoCo simulator [20], which is a physics-based simulator, using two environments with unknown model dynamics: *Walker2d* and *Hopper*. The goal of the hopper problem is to make a three degree of freedom mono pedal move forward as fast as possible without falling. The goal of the walker2d problem is to move forward as quickly as possible with a bipedal walker constrained to the plane without falling down and pitching the torso too far forward or backward. In both problem, the underlying reward is defined as the sum of survival rewards, forward moving speed, and negative control quadratic rewards. More details can be found in [21].

We verify that sub-optimal demonstrations are useful to train dynamic networks. We also compare the performance of the proposed method with IOCfOD to prove the sample efficiency of the proposed method.

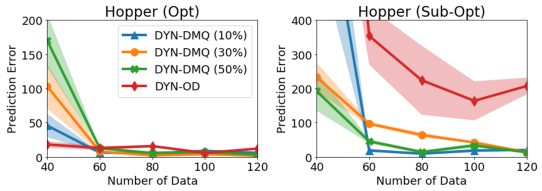
1) Generation of Sub-Optimal Demonstrations: While the wide range of sub-optimal policies can be used to generate sub-optimal demonstrations, we would like to select more meaningful one which satisfies two conditions: (1) sub-optimal demonstrations should be distributed near the catastrophic states and (2) provide the information about which control leads to a failure of a task.

To generate such sub-optimal demonstrations, we first sample an episode by controlling an agent using a random policy and a failed episode can be obtained. In both hopper and walker2d problems, the failure of a task is defined as the falling of an agent. Then, the most informative state is the final state of the failed episode since it is the state which should be avoided. In this regards, we segment the last of 5 state action pairs including the final state from the sampled demonstration. Since we generate sub-optimal demonstrations near the failed state, we can expect that using the generated demonstrations effectively provide the information about the failure case.

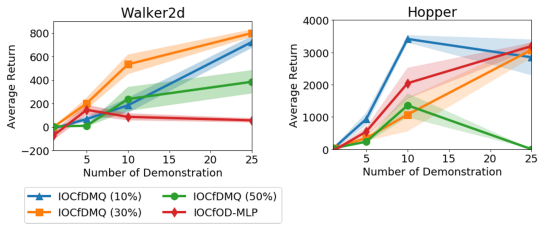
2) Learning Dynamic Model: The state, action, and next state pairs generated by an expert are timely correlated. In this regards, when we train the dynamic network using optimal demonstrations, the correlated data lead to a poor generalization performance of the resulting dynamic model. However, by mixing optimal and sub-optimal demonstrations, the entire training data become more uncorrelated. To prove the effect of sub-optimal demonstrations, we prepare four types of training demonstrations. The first one consists of only optimal demonstrations and the others contain both optimal and sub-optimal demonstrations with three different ratios: $10\%, 30\%$, and 50%. We train a dynamic model with different numbers of state-action pairs $(40, 60, 80, 100, 120)$.

To investigate the influence of sub-optimal demonstrations to the generalization performance of a dynamic network, we prepare 100 optimal state action pairs and 100 sub-optimal state action pairs as test data of the trained dynamic networks, which are different from the training demonstrations. The trained network is tested on the task of predicting the next state. The prediction error is measured by the $l2$ distance. The average and standard deviation about the prediction error with five different random seeds are shown in Figure 2(a).

As we can expect, the dynamic models trained with sub-optimal demonstrations generally shows lower prediction error for sub-optimal data. However, it is also observed that using sub-optimal demonstrations also helps to predict the optimal demonstrations. In Figure 2(a), when the number of data is greater than 60, the prediction error of the dynamic model trained with sub-optimal demonstrations is generally lower than that trained with only optimal demonstrations. This result supports that using both optimal and sub-optimal demonstrations improve the generalization performance of the trained dynamic model. Since the proposed method highly depends on $G_\nu(x_t)$, the generalization performance of $G_\nu(x_t)$ influences the average return of the resulting policy.



(a) Prediction Error of Dynamic Model



(b) Average Return in MuJoCo

Fig. 2: The number in the parentheses shows the ratio of the number of sub-optimal demonstrations. (a) Prediction error. Opt is the test result on the optimal demos, Sub-Opt is the test result on the sub-optimal demos. (b) Average returns of the trained policy on the hopper and walker2d problems.

3) Inverse Optimal Control: We apply both IOCfDMQ and IOCfOD to different numbers of mixed demonstrations with different ratio: $10\%, 30\%$, and 50% and measure the average returns over 100 consecutive episodes using the resulting policy. The five simulations are conducted with different random seeds. The results are shown in Figure 2(b). In both problems, the proposed method outperforms IOCfOD. IOCfDMQ with 5% sub-optimal demonstrations show the best performance and the IOCfDMQs with other ratios are worse than IOCfOD in the hopper problem. In the walker2d problem, IOCfDMQ generally outperforms IOCfOD. The hyperparameters of the both methods are found by the brute force search. The first reason why IOCfDMQ shows better performance than IOCfOD is that the dynamic model trained with the demonstrations with mixed quality shows better generalization performance. Furthermore, we believe that this result is mainly caused by the regularization effect which is already shown in the pendulum problem. One interesting observation is that IOCfDMQ with 30% sub-optimal demonstrations shows poor performance than IOCfOD in both problems. We believe that this performance drop is caused due to the insufficient number of optimal demonstrations.

VI. Conclusion

We have proposed an inverse optimal control method using demonstrations with mixed qualities. The proposed method utilizes the benefits of sub-optimal demonstrations which give information about *what not to do* and provides training data near the states unvisited by optimal demonstrations. In experiments, the proposed method outperforms the original IOC algorithm, which uses only optimal demonstrations, with a smaller number of demonstrations. As a result, we can conclude that using sub-optimal demonstrations has an effect of alleviating the overfitting issue and is also beneficial to the generalization performance of the dynamic network when a model is not available.

References

[1] P. Abbeel, A. Coates, and A. Y. Ng, "Autonomous helicopter aerobatics through apprenticeship learning," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1608–1639, 2010.

[2] D. Vasquez, B. Okal, and K. Arras, "Inverse reinforcement learning algorithms and features for robot navigation in crowds: an experimental comparison," in *Proc. of the International Conference on Intelligent Robots and Systems*, 2014, pp. 1341–1346.

[3] B. Okal and K. O. Arras, "Learning socially normative robot navigation behaviors with bayesian inverse reinforcement learning," in *Proc. of the International Conference on Robotics and Automation*. IEEE, 2016, pp. 2889–2895.

[4] C. Finn, S. Levine, and P. Abbeel, "Guided cost learning: Deep inverse optimal control via policy optimization," in *Proc. of the International Conference on Machine Learning*, Jun 2016, pp. 49–58.

[5] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proc. of the international conference on artificial intelligence and statistics*, Apr 2011, pp. 627–635.

[6] K. Shiarlis, J. V. Messias, and S. Whiteson, "Inverse reinforcement learning from failure," in *Proc. of the International Conference on Autonomous Agents & Multiagent Systems*, May 2016, pp. 1060–1068.

[7] K. Lee, S. Choi, and S. Oh, "Inverse reinforcement learning with leveraged gaussian processes," in *Proc. of the International Conference on Intelligent Robots and Systems*, Oct 2016, pp. 3907–3912.

[8] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning." in *Proc. of the Twenty-Third AAAI Conference on Artificial Intelligence*. AAAI Press, Jul 2008, pp. 1433–1438.

[9] N. D. Ratliff, J. A. Bagnell, and M. Zinkevich, "Maximum margin planning," in *Proc. of the International Conference on Machine Learning*, Jun 2006, pp. 729–736.

[10] S. Levine, Z. Popovic, and V. Koltun, "Nonlinear inverse reinforcement learning with gaussian processes," in *Advances in Neural Information Processing Systems*, Dec 2011, pp. 19–27.

[11] N. Aghasadeghi and T. Bretl, "Maximum entropy inverse reinforcement learning in continuous state spaces with path integrals," in *Proc. of the International Conference on Intelligent Robots and Systems*, Sep 2011, pp. 1561–1566.

[12] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Advances in Neural Information Processing Systems*, Dec 2016, pp. 4565–4573.

[13] J. Audiffren, M. Valko, A. Lazaric, and M. Ghavamzadeh, "Maximum entropy semi-supervised inverse reinforcement learning," in *Proc. of the International Joint Conference on Artificial Intelligence*, Jul 2015, pp. 3315–3321.

[14] J. Zheng, S. Liu, and L. M. Ni, "Robust bayesian inverse reinforcement learning with sparse behavior noise," in *Proc. of the AAAI International Conference on Artificial Intelligence*, Jul 2014, pp. 2198–2205.

[15] M. Valko, M. Ghavamzadeh, and A. Lazaric, "Semi-supervised apprenticeship learning," in *Proc. of the European Workshop on Reinforcement Learning*, Jun 2012, pp. 131–142.

[16] W. Li, E. Todorov, and D. Liu, "Inverse optimality design for biological movement systems," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 9662–9667, 2011.

[17] E. Pauwels, D. Henrion, and J. Lasserre, "Inverse optimal control with polynomial optimization," in *Proc. of the International Conference on Decision and Control*, Dec 2014, pp. 5581–5586.

[18] A.-S. Puydupin-Jamin, M. Johnson, and T. Bretl, "A convex approach to inverse optimal control and its application to modeling human locomotion," in *Proc. of the International Conference on Robotics and Automation*, May 2012, pp. 531–536.

[19] H. Ravanbakhsh and S. Sankaranarayanan, "Learning lyapunov (potential) functions from counterexamples and demonstrations," in *Robotics: Science and Systems*, Jul 2017.

[20] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *Proc. of the International Conference on Intelligent Robots and Systems*, Oct 2012.

[21] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. M. O. Heess, T. Erez, Y. Tassa, D. Silver, and D. P. Wierstra, "Continuous control with deep reinforcement learning," Jan 2017, uS Patent App. 15/217,758.