2017 IEEE International Conference on Robotics and Automation (ICRA)
Singapore, May 29 - June 3, 2017

Apprenticeship Learning in an Incompatible Feature Space

Gakuto Masuyama[1] and Kazunori Umeda[1]

Abstract— **This study presents a novel apprenticeship learning method to enable a learner to utilize demonstrations observed in an incompatible feature space. It is assumed that an expert and a learner follow non-identical Markov decision processes (MDPs), and a mapping function is estimated to obtain feature expectation of the demonstrations in an agent space. A conditional density estimation technique is used to represent the feature expectation in closed-form. The proposed method is useful because it is expected to alleviate intractable processes to explicitly specify correspondence of heterogeneous MDPs for apprenticeship learning. Additionally, the method does not require any sampling method to approximate integrals over an agent feature space. A simulation is used to demonstrate the validity of the proposed method in three domains in which it is not possible to directly compare the features of the expert and learner.**

I. INTRODUCTION

Reinforcement Learning (RL) [1] is actively studied as a tool for the development of a fully autonomous and adaptive robot control system. Specifically, RL is a general framework to learn control policy from collected experience obtained from interaction between a robot and its environment. The objective of RL involves maximizing the expected cumulative reward signal that is received by the robot as a consequence of each decision made by the robot. The effectiveness of RL for robotic control problems is demonstrated in large studies [2], [3], [4].

Although RL has favorable characteristics with respect to robotics, open problems persist for practical use scenarios. A fundamental problem involves the design of a reward function that encodes a task given to the robot. Typically, a reward function is specified by manual coding. The design of the reward function strongly affects learning performance of the robot. Therefore, the provision of an appropriate reward function is a cumbersome task for a designer. Extant research proposed potential solutions to the fore-mentioned problem by developing Inverse Reinforcement Learning (IRL) and Apprenticeship Learning (AL) [5], [6], [7]. RL learns policy based on a reward function. Conversely, IRL estimates a reward function based on observed demonstrations of experts in which the experts are assumed to know (nearly) optimal policy for the given task. The objective of an AL framework is to recover the policy of an expert. In order to recover the policy, an AL agent first estimates a reward function using IRL and then learns policy by using a forward RL procedure with the estimated reward function. AL has particularly useful properties for robotic control problem, and we consider an extremely useful advantage of using AL is its generalization capability.

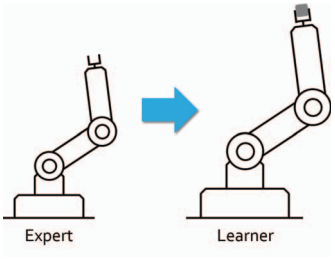
In [6], Abbeel stated that a reward function is "the most succinct, robust, and transferable definition of the task." Direct imitation learning methods mimic policy. In contrast, AL abstracts the demonstration as a reward function (that is analogous to the intention of an expert). The policy is then crystallized using forward RL based on a robot's experience. Hence, the policy is re-constructed via the experience of the robot instead of observations of the expert itself. The property leads to remarkable capability with AL since it enables the robot to generalize its policy to regions in which demonstrations are not observed. Currently, the fore-mentioned advantage is enhanced as an extension to incomplete and noisy demonstrations [8], [9].

In this study, another challenge is considered to generalize the AL framework. It is assumed that a feature observed from an expert and a feature used to represent the estimated reward function are not identical. In this case, it is not possible to directly apply most IRL procedures such as *feature matching* [6]. Discrepancy in features often appears if dynamics or environment of an expert and a robot are not identical. This is almost always applicable with respect to the imitation learning of human behavior by robots. Generally, a system designer is required to create a handcrafted common feature to overcome a discrepancy by intuition. However, this potentially limits AL applications to only carefully pre-specified tasks.

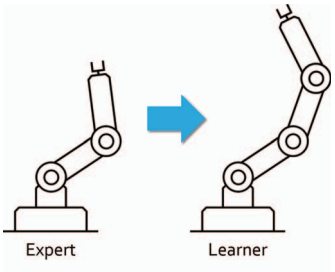
Fig. 1 depicts an illustrative example of discrepancy of features that arises from a difference in body structures of the robots. As is widely known, a bottleneck of RL in robotics corresponds to learning speed. This is mainly due to the difficulty in collecting relevant samples with respect to a specific robot and a specific environment. Thus, a reward function is important in enabling efficient exploration. The present study focuses on eliminating the fore-mentioned AL limitation.

A novel AL procedure is proposed in this study to alleviate the limitation. A conditional density estimation technique is utilized for the feature matching framework. Feature expectation of demonstrations is represented in a learner's feature space by estimating the conditional density of an expert's feature in the learner's feature space. The reward function is then estimated by using IRL. Finally, forward RL is used to learn policy from the estimated reward function. The validity of the proposed method is verified through simulations in three domains. The results indicated that the proposed method is useful because it is simple and computationally efficient. Hereinafter, the term "agent" is

[1]Authors are with the Department of Precision Mechanics, Faculty of Science and Engineering, Chuo University, 1-13-27 Kasuga, Bunkyo-ku, Tokyo 113-8551, Japan. {masuyama, umeda} at mech.chuo-u.ac.jp



(a) Generalization of policy in AL



(b) Discrepancy in feature space

Fig. 1: The objective of robots involves controlling joint angles as specified in the figure. (a) A learner possesses a scaled-up body of an expert and holds an object with its end effector. However, an expert can transfer its reward function if they share a feature with the learner such as *e.g.* angular information of corresponding joints. The learner then can convert the reward function into its localized policy. (b) An illustrative example of discrepancy in features due to differences in the body structure. It is not possible to apply IRL without a carefully designed relevant feature.

used to denote a learner of AL instead of a robot.

II. PRELIMINARIES

It is assumed that the problem in the study is formalized as a Markov decision process (MDP) that is represented by a tuple $(S, A, T, R, d_0, \gamma)$. Specifically, S denotes a set of states; A denotes a set of actions; $T(s, a, s') = Pr(s_{t+1} = s' | s_t = s, a_t = a)$ denotes a transition probability from a state $s \in S$ to the next state $s' \in S$ under action a; R denotes the reward function; d_0 denotes an initial state distribution; and $\gamma \in [0, 1)$ denotes a discount rate. A stochastic policy $\pi(s, a) : S \times A \mapsto [0, 1]$ assigns a probability of selecting an action a in state s. The objective of RL is to obtain an optimal policy π^* that maximizes the expected return $E[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) | d_0, \pi, T]$, i.e., the value function $V^\pi(s)$.

The objective of IRL is to estimate a reward function based on expert demonstrations. Therefore, MDP without a reward function MDP$\backslash R$ is assumed. The reward function is represented by a linear combination of a parameter vector $\boldsymbol{\theta}$ and a feature vector. A feature vector of the expert is denoted by $\boldsymbol{x} \in \mathbb{R}^{d_X}$, and a feature vector of the agent is denoted by $\boldsymbol{y} \in \mathbb{R}^{d_Y}$. The estimate of the reward function is then represented as follows: $\hat{R}(s, a) := \sum_{i=1}^{d_Y} \theta_i y_i(s, a)$, where y_i and θ_i denote the ith element of $\boldsymbol{y}$ and $\boldsymbol{\theta}$, respectively. The expected feature under policy π is denoted as follows: $y_i^\pi = E[\sum_{t=0}^{\infty} \gamma^t y_i(s_t, a_t) | d_0, \pi, T]$. This notation simplifies the value function; $V^\pi(s) = \sum_{i=1}^{d_Y} \theta_i y_i^\pi(s, a)$.

III. METHOD

It is assumed that the reward function is estimated by IRL based on feature matching, which is potentially the most common framework used in extant literature. First, conditional density estimator of $\boldsymbol{y}$ given $\boldsymbol{x}$ is trained using paired samples. Second, feature expectation in an agent's feature space is estimated using demonstrations observed in an expert's feature space. The feature expectation is used to estimate the reward function by IRL. Finally, policy is recovered from the estimated reward function by implementing forward RL.

A. Feature Matching in AL

A summary of feature matching in AL is first presented. As stated above, it is assumed that a reward function is represented by a linear combination of $\boldsymbol{\theta}$ and $\boldsymbol{x}$ and that the value function can then be represented by $\boldsymbol{\theta}$ and an expectation of features. A constraint is imposed in feature matching as follows:

$$\|\boldsymbol{\theta}^T \boldsymbol{x}^{\pi_E} - \boldsymbol{\theta}^T \boldsymbol{x}^{\pi}\| \leq \varepsilon, \tag{1}$$

where $\boldsymbol{x}^{\pi_E}$ denotes the expected feature observed from the policy of an expert π_E, and $\varepsilon \in \mathbb{R}_+$ denotes a margin between the two value functions. $\boldsymbol{x}^{\pi_E}$ is often given in terms of an empirical estimate. The parameter vector $\boldsymbol{\theta}$ is optimized under the constraint in eq.(1) to match the observed behavior of the expert (π_E) and the agent (π). The formulation evidently relies on the assumption that the expert and agent share the same feature space. The assumption is relaxed, and it is assumed that the feature of the expert $\boldsymbol{x} \in X$ is not compatible with that of the agent $\boldsymbol{y} \in Y$.

One of the simplest approaches to handle the incompatibility of features involves mapping the feature of the expert $\boldsymbol{x}$ into the feature space of the agent Y. Our objective then is to obtain $\hat{\boldsymbol{y}} := \underset{p(\boldsymbol{y})}{E}[\boldsymbol{y}]$ from a set of observed demonstrations $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$.

B. Conditional Density Estimation

It is not possible to obtain the feature expectation in an agent feature space $\hat{\boldsymbol{y}}$ in the problem stated in the present study because the demonstrations are given in the expert feature space. As opposed to using an empirical estimate of an expert's feature expectation as a usual IRL, it is marginalized as follows:

$$\begin{aligned} \hat{\boldsymbol{y}} &= \int \boldsymbol{y} p(\boldsymbol{y}) d\boldsymbol{y} \\ &= \iint \boldsymbol{y} p(\boldsymbol{y}|\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x} d\boldsymbol{y}. \end{aligned} \tag{2}$$

Additionally, $p(\boldsymbol{y}|\boldsymbol{x})$ is estimated by using some paired samples $\{\boldsymbol{x}_i^c, \boldsymbol{y}_i^c\}_{i=1}^{n_c}$. It is assumed that the paired samples

are given a priori and assumed as relevant irrespective of the given tasks [1].

A Least-Squares Conditional Density Estimation (LSCDE) [10] is used as an estimator of conditional density $p(\boldsymbol{y}|\boldsymbol{x})$. LSCDE does not directly estimate conditional density. It estimates density ratio $\rho(\boldsymbol{x},\boldsymbol{y})$ of $p(\boldsymbol{x},\boldsymbol{y})$ to $p(\boldsymbol{x})$ as opposed to naively estimating $p(\boldsymbol{x},\boldsymbol{y})$ and $p(\boldsymbol{x})$ as follows:

$$\begin{aligned}\rho(\boldsymbol{x},\boldsymbol{y}) &:= \frac{p(\boldsymbol{x},\boldsymbol{y})}{p(\boldsymbol{x})} \\ &= p(\boldsymbol{y}|\boldsymbol{x}).\end{aligned} \tag{3}$$

The estimation of the density ratio enables the avoidance of large errors in the ratio obtained from a naively estimated $p(\boldsymbol{x},\boldsymbol{y})$ and $p(\boldsymbol{x})$. As is known, naive standard density estimation techniques [11] tend to be unreliable and especially in the case of a high dimensional problem. The conditional density is approximated by a linear combination of parameters $\boldsymbol{\alpha} \in \mathbb{R}^b$ and basis functions $\boldsymbol{\phi} : X \times Y \to \mathbb{R}^b$ as follows:

$$\hat{\rho}_{\boldsymbol{\alpha}}(\boldsymbol{x},\boldsymbol{y}) := \boldsymbol{\alpha}^T \boldsymbol{\phi}(\boldsymbol{x},\boldsymbol{y}). \tag{4}$$

It should be noted that $\hat{\rho}_{\boldsymbol{\alpha}}$ corresponds to the value that is not normalized. Please refer to [10] for more details on LSCDE. Next, an estimate of the feature expectation is derived in an agent feature space by using LSCDE.

C. Feature Expectation in Agent Feature Space

The estimated density ratio $\hat{\rho}_{\boldsymbol{\alpha}}$ is used to approximate the feature expectation $\hat{\boldsymbol{y}}$ in eq.(2). A Gaussian kernel is assumed, and its jth element is defined as follows:

$$\phi_j(\boldsymbol{x},\boldsymbol{y}) := \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{u}_j\|^2}{2\sigma^2}\right)\exp\left(-\frac{\|\boldsymbol{y}-\boldsymbol{v}_j\|^2}{2\sigma^2}\right), \tag{5}$$

where $\boldsymbol{u}_j \in X$ and $\boldsymbol{v}_j \in Y$ denote center points, and $\sigma \in \mathbb{R}_+$ denotes Gaussian width.

The probability density of $\boldsymbol{y}$ is then represented as follows:

$$\begin{aligned}p(\boldsymbol{y}) &= \int p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} \\ &\approx \int \frac{\hat{\rho}_{\boldsymbol{\alpha}}(\boldsymbol{x},\boldsymbol{y})}{\int \hat{\rho}_{\boldsymbol{\alpha}}(\boldsymbol{x},\boldsymbol{y}')d\boldsymbol{y}'}p(\boldsymbol{x})d\boldsymbol{x} \\ &\approx \frac{1}{n}\sum_{i=1}^{n}\frac{\hat{\rho}_{\boldsymbol{\alpha}}(\boldsymbol{x}_i,\boldsymbol{y})}{\int \hat{\rho}_{\boldsymbol{\alpha}}(\boldsymbol{x}_i,\boldsymbol{y}')d\boldsymbol{y}'},\end{aligned} \tag{6}$$

where $\{\boldsymbol{x}_i\}_{i=1}^n$ denotes a demonstration by an expert. The integral in a normalizing constant $Z(\boldsymbol{x}_i) := \int \hat{\rho}_{\boldsymbol{\alpha}}(\boldsymbol{x}_i,\boldsymbol{y}')d\boldsymbol{y}'$ can be analytically solved. The feature expectation in an agent feature space is then obtained using eq.(6) as follows:

$$\begin{aligned}\hat{\boldsymbol{y}} &= \int \boldsymbol{y}p(\boldsymbol{y})d\boldsymbol{y} \\ &\approx \frac{1}{n}\sum_{i=1}^{n}\frac{1}{Z(\boldsymbol{x}_i)}\int \boldsymbol{y}\boldsymbol{\alpha}^T\boldsymbol{\phi}(\boldsymbol{x}_i,\boldsymbol{y})d\boldsymbol{y} \\ &= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{b}\frac{1}{Z(\boldsymbol{x}_i)}\exp\left(-\frac{\|\boldsymbol{x}_i-\boldsymbol{u}_j\|^2}{2\sigma^2}\right) \\ &\qquad \int \boldsymbol{y}\alpha_j \exp\left(-\frac{\|\boldsymbol{y}-\boldsymbol{v}_j\|^2}{2\sigma^2}\right)d\boldsymbol{y} \\ &= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{b}\frac{\alpha_j \exp\left(-\frac{\|\boldsymbol{x}_i-\boldsymbol{u}_j\|^2}{2\sigma^2}\right)}{\sum_{k=1}^{b}\alpha_k \exp\left(-\frac{\|\boldsymbol{x}_i-\boldsymbol{u}_k\|^2}{2\sigma^2}\right)}\boldsymbol{v}_j.\end{aligned} \tag{7}$$

Thus, a closed form solution as shown in eq.(7) is obtained in a case in which the Gaussian kernel is chosen. The Gaussian kernel in an expert feature space is weighted by α_j and composes a soft-max like activation function. Each center point of the basis function in the agent feature space $\boldsymbol{v}_j$ is then activated by $\boldsymbol{x}_i$, which is observed from a demonstration of an expert. Finally, the center points are averaged over all demonstrated feature samples. It is possible to efficiently compute the feature expectation in eq.(7) because it does not require a sampling method to approximate integrals over agent feature space.

D. Recovering Policy

An arbitrary IRL method can be implemented for reward estimation if it relies on $\hat{\boldsymbol{y}}$ or $\hat{p}(\boldsymbol{y})$. In this study, Relative Entropy Inverse Reinforcement Learning (REIRL) [12] is used. REIRL is model-free and does not require an iterative forward RL procedure. Any RL is also applicable, and classical value iteration [1] is used in the following simulations. The AL proposed in the study is favorable because it involves a fairly straightforward structure and is easy to implement. An overview of the entire procedure is shown in Algorithm 1 as follows:

Algorithm 1 Procedure of the proposed AL

1: **input**: $\{(\boldsymbol{x}_i^c,\boldsymbol{y}_i^c)\}_{i=1}^{n_c}$, $\{\boldsymbol{x}_i\}_{i=1}^{n}$
2: **output**: π
3: $(\boldsymbol{\alpha},\boldsymbol{u},\boldsymbol{v},\sigma) = \text{LSCDE}(\boldsymbol{x}^c,\boldsymbol{y}^c)$ # train conditional density estimator
4: Estimate $\hat{\boldsymbol{y}}$ by eq.(7) # mean for $\{\boldsymbol{x}_i\}_{i=1}^{n}$
5: Estimate reward function IRL
6: Learn policy by forward RL

IV. SIMULATION

Simulations are performed in three domains to verify the validity of proposed AL. A simulation corresponds to *3D to 2D grid world domain* in which an expert is in a 3D grid world and agent is in a 2D grid world. The correspondence between X and Y is clear in the above scenario, and thus the result can be quantitatively evaluated. Another simulation

[1]The fore-mentioned assumption could be rather strong. Although this is not discussed with respect to a specific application in this study, the prospect for a robotic control problem is described in IV-D.

corresponds to simplified *link arms domain* in which an expert is a 2-link jointed arm and an agent is a 3-link jointed arm, which result in different body structures. In this scenario, it is not possible to clearly illustrate the correspondence between X and Y. Thus, empirical results are provided for the discussion to apply the proposed method to a heterogeneous real system. Finally, a demonstration is performed to extend the results in *Mountain-car to Pendulum domain* in a continuous state and action space. The proposed method is also compared with a method that uses linear regression to obtain inter-feature mapping. The final purpose involves controlling a robot (possibly high dimensional) in which it is difficult to manually design the reward function. However, a principled method to address the corresponding points is required for a fair evaluation of the method presented in this study. This continues to be an open problem, and thus, a straightforward domain is selected in the study. The problem of the corresponding point is discussed in section IV-D.

A. 3D to 2D Grid World domain

1) Settings: The expert / agent is in a 3D / 2D grid world in which its features are represented as $\boldsymbol{x} = (x_e, y_e, z_e) \in [1, 15]^3$ / $\boldsymbol{y} = (x_a, y_a) \in [1, 15]^2$. It is assumed that (x_a, y_a) corresponds to a normally projected point of (x_e, y_e, z_e) on a $x_a y_a$-plane. Thus, the objective of the agent involves learning the relationship $(x_a, y_a) = (x_e, y_e)$ and estimating an underlying reward function to recover the corresponding policy.

Each axis was equally divided by 15 for discretization purposes. Available actions included moving $[-1, 0, 1]$ for every axis, i.e. the expert and agent involved 3^3 and 3^2 actions, respectively. The true reward function for expert yielded 1 in the goal state $\boldsymbol{x}^g$ and 0 otherwise. The demonstrations involved 30 trajectories. Each trajectory was sampled by using an optimal policy with uniformly distributed random initial states. The number of paired samples corresponded to 100. The paired samples were obtained from $\{\boldsymbol{y}_i\}_{i=1}^{100}$, which were drawn from a uniform distribution over an agent feature space.

The number of kernels corresponded to 100; and a parameter for regularization in LSCDE and σ were selected from log-spaced values by five-fold cross-validation. Additionally, $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ were randomly selected from the given paired samples.

Parameters of the REIRL were determined as the discount rate $\gamma = 0.98$, and the margin for feature matching $\varepsilon = 0.05$.

We performed the simulations under the following three conditions: a) $\boldsymbol{x}^g = (7, 7, 7)$; b) $\boldsymbol{x}^g = (12, 12, 3)$; and c) $\boldsymbol{x}^g = (12, 12, 12)$. Furthermore, with respect to upper and lower baselines, the following were executed: 1) REIRL given optimal trajectories in an agent feature space and 2) REIRL given randomly sampled trajectories. In 2), the trajectories were sampled by a uniform distribution over action space. Thus, the result should indicate the lower bound of performance. Each condition was tested 20 times.

TABLE I: Results of policy loss $L[10^{-2}]$ for each $\hat{\boldsymbol{x}}^g$.

	(7,7,7)	(12,12,3)	(12,12,12)
Proposed (mean)	1.62	1.45	1.83
Proposed (sd)	0.73	0.64	0.86
Optimal (mean)	0	0	0
Optimal (sd)	0	0	0
Random (mean)	5.16	7.38	7.75
Random (sd)	1.92	2.91	2.43

TABLE II: Results of terminal state for each $\hat{\boldsymbol{x}}^g$. x and y represent horizontal axis and vertical axis, respectively, in the agent feature space(see Fig. 2).

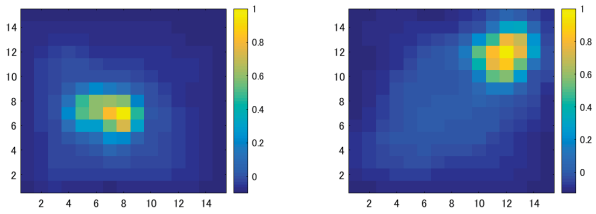
	$(7,7,7)$		$(12,12,3)$		$(12,12,12)$	
	x	y	x	y	x	y
Proposed (mean)	6.9	7.0	11.8	11.8	11.5	11.7
Proposed (sd)	1.3	0.8	0.9	1.1	1.4	1.2
Optimal (mean)	7.0	7.0	12.0	12.0	12.0	12.0
Optimal (sd)	0.0	0.0	0.0	0.0	0.0	0.0
Random (mean)	7.6	8.2	8.0	7.4	8.0	5.7
Random (sd)	2.8	3.8	4.9	4.3	4.2	4.2

2) Results: The variant of policy loss L [13] (we call simply policy loss hereinafter) is evaluated, and a state in which the reward function corresponds to the highest value $\hat{\boldsymbol{y}}^g$ is given. The policy loss represents a similarity between the recovered policy and optimal policy in the simulation. The difference from the original policy loss involves using a value function learned from an estimated reward function. The state corresponding to the highest reward coincides with a terminal state.

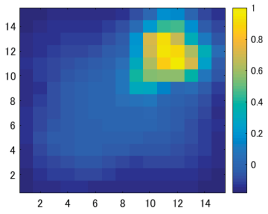
Table I shows results with respect to policy loss. It should be noted that each value function is normalized. Table II shows results with respect to the terminal state. The results indicate that it is possible to recover an almost perfect policy given access to demonstrations in the agent space. The terminal states had a tendency to assume smaller values than that in the true terminal state in case of $(12, 12, 3)$ and $(12, 12, 12)$. The goal states were positioned near the corner of grid world in the above two conditions, and the demonstrations were sampled from randomly chosen initial states. Thus, the distribution of features involved a spatial bias. By considering the above fact, the proposed method is capable of recovering an appropriate policy even if the feature spaces of the expert and agent differ from each other.

Fig. 2 depicts the averaged reward function estimated by the proposed method. It is observed that the estimated reward function assumes the highest value at the corresponding goal state in each condition.

The proposed method is also tested while varying the number of corresponding points. Table III shows the mean and standard deviation of policy loss in which $\boldsymbol{x}^g = (12, 12, 12)$. The result indicates that the method can reliably estimate the reward function as the number of corresponding points



(a) $\boldsymbol{x}^g = (7, 7, 7)$ (b) $\boldsymbol{x}^g = (12, 12, 3)$



(c) $\boldsymbol{x}^g = (12, 12, 12)$

Fig. 2: Mean reward function.

TABLE III: Results of policy loss $L[10^{-2}]$ for the varying number of corresponding points.

# of points	10	25	50	100	150	200
mean	4.49	2.84	1.35	1.59	1.54	1.31
sd	1.53	1.78	0.76	0.69	0.66	0.68

increases. The number of states corresponds to 3375 in the 3D space and 225 in the 2D space. Given that the points were sampled by a uniform distribution, the proposed method appears to possess a fairly good generalization ability.

B. Link Arms domain

1) Settings: This is followed by investigating the manner in which the difference in body structure of the expert and agent affect the performance of the proposed method. The expert / agent corresponds to a simplified planar 2 / 3-link arm. All link lengths correspond to 1; and thus there are differences in the degree of freedom and complete length between the expert and the agent. The features correspond to the end-point position of each link, and the states correspond to tuples of joint angles.

Each joint angle was equally divided by 16 for discretization purposes. Each link was able to independently execute actions to stay or transit to an adjacent joint angle. However, in the study, a transition was prohibited to a state in which jointed links conflict with each other. It should be noted that it may not represent explicit correspondence between the expert and agent in the domain. Therefore, paired samples were generated by an intuitively straightforward correspondence between the expert and agent as shown in Fig. 3. Three postures are depicted in the figure. Each posture in the left and right figures is denoted by color. The basic three postures were rotated around the origin, and 3×16 paired samples

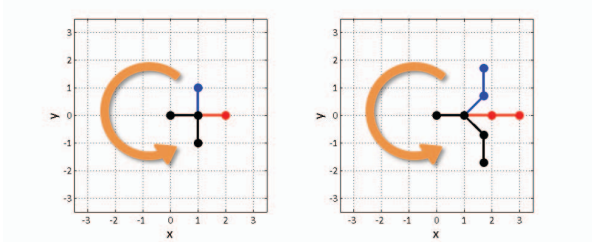


Fig. 3: The given paired samples. Blue, red, and black pairs are rotated around the origin.

TABLE IV: Results of the end-point position for each link.

	1st		2nd		3rd	
	x	y	x	y	x	y
Proposed (mean)	0.04	0.95	0.05	1.84	0.13	2.72
Proposed (sd)	0.32	0.07	0.43	0.12	0.42	0.16
Optimal (mean)	0.00	1.00	0.00	2.00	0.00	3.00
Optimal (sd)	0.00	0.00	0.00	0.00	0.00	0.00
Random (mean)	0.10	-0.15	0.28	-0.35	0.13	-0.13
Random (sd)	0.76	0.67	1.24	1.12	1.65	1.24

were obtained. The other settings were the same as those in section IV-A.

2) Results: The following two tasks were assigned to the expert: a) true reward function of the expert corresponding to 1 at $(\pi/2, \pi/2)$, and b) $(\pi/4, \pi/2)$, otherwise 0. A goal in condition a) coincides with one of the paired samples. Therefore, the simulations involve investigating the terminal end-point position of links for a) and terminal posture by recovered policy for b).

The end-point position in condition a) is shown in Table IV. In order to obtain the terminal states, estimated policy with initial states $(-\pi/2, -\pi/2, -\pi/2)$ was used. An almost perfect terminal state was recovered by using a direct IRL procedure with optimal demonstrations. It is confirmed that the proposed method approximately estimated an accurate reward function although there were differences in the feature spaces of the expert and agent in terms of dimensionality and body length. The number of paired samples was fewer when compared with that in a grid world domain. Thus, the decrease in performance of the proposed method over *Optimal* case is mainly due to the number and distribution of paired samples.

Fig. 4 illustrates the averaged terminal posture in condition b). The agent appears to appropriately interpolate among the paired samples to achieve a posture analogous to that of the expert.

C. Mountain-car to Pendulum domain

1) Settings: Another simulation is performed in which the reward function is transferred from demonstrations in mountain-car problem to a pendulum swing-up problem. The state and action space are continuous in both problems. In

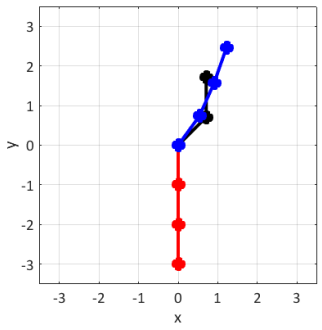


Fig. 4: Mean terminal state obtained by the proposed method. Black lines represent the terminal state of the expert; red lines represent an initial state of the agent; and blue lines represent the averaged terminal state of the agent.

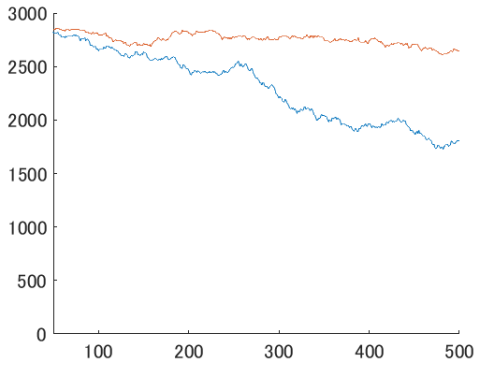


Fig. 5: Time steps over episodes. Blue line represents learning curve of the proposed method; red line represents learning curve using linear regression.

this domain, it is difficult to obtain a correspondence in the feature space of the expert and agent. However, they should both possess cyclic movements to achieve given tasks. Therefore, it is tested as to whether the proposed method can specify the implicit correspondence. This is assumed as the case in realistic robotic control problems.

In this study, a heuristic approach was adopted to sample the corresponding points. First, the start and terminal state of trajectories were specified (they correspond to the start and terminal state that satisfy the terminal conditions of tasks). Second, trajectories that passed the start and terminal state were sampled based on arbitrary policy. A Gaussian exploration policy was used for the agent (mountain-car), and one of the demonstrations was used for the expert (pendulumn). Finally, the two trajectories were normalized with respect to the time step, and the corresponding points were then sampled from interpolated trajectories. For comparison purposes, linear regression was implemented using the corresponding points to obtain maps on the feature space.

Deterministic policy gradient [14] with RBF features was used to obtain policy from the estimated reward function. The number of trials corresponded to 5. The other settings were the same as those in section IV-A.

2) Results: Fig. 5 depicts the averaged learning curves over 5 trials in which the horizontal axis represents an episode, and the vertical axis represents time steps in an episode. The blue area corresponds to the proposed method; and the red area corresponds to the linear regression case. Both learning curves are smoothed by a moving average filter with a window size of 50. It should be noted that the results were obtained from a stochastic Gaussian policy with a fixed variance. The findings confirmed that the proposed method learned a successful deterministic policy across all trials.

Linear regressed mapping improved the policy slowly. However, the proposed method found a reasonable policy faster. The results indicate that the proposed method is at least partially successful in mapping implicit correspondence between the two cyclic movements. It is considered that one of the advantages of the proposed method involves utilizing the generalization ability of RL even if the mapping is not completely thorough with respect to the complete feature space.

D. Discussion

An intuitive method to quantitatively evaluate the results in Fig. 4, 5 is not evident because it is not possible to define the correspondence between the expert and the agent unless a clear criterion is provided. This may be a strong limitation of the proposed method. Although it is not possible for the proposed method to provide an ideal transfer for the system designer [2] , this possibly reproduces "meaningful" behavior. Thus, we consider to combine the proposed method with active learning [15] to refine the rough behavior. This framework may reduce supervision cost. Another expected benefit relates to data efficiency as noted in section I. The proposed method aids in accelerating the learning of RL agents, and especially in an early phase of learning. It is expected that this functionality is fundamental for robot control by RL.

There continues to be scope for improvement in the method: a) large difference in the dimensionality of feature space, and b) a procedure to obtain paired samples in a real robot control problem. The former problem potentially increases the necessary amount of paired samples. The utilization of dimensionality reduction technique could alleviate the problem. The latter is more fundamental in practice. There are many possible scenarios depending on applications. A method involves designing a baseline task to generate paired "trajectories" rather than "points" (this is performed in a relatively heuristic manner in the Mountain-car to Pendulum domain). The most reliable approach corresponds to kinesthetic teaching. Another method involves extending the proposed framework to multi-agent systems. A known mediator can be used to estimate the intention of another subject.

[2]This is inevitable because the "ideal" depends on our subjective view in case we consider systems composed of heterogeneous agents including various robots and humans.

The problem addressed in this study may corresponds to *correspondence problem*. As stated in a previous study, with respect to the transfer of skill by observation, behavior must already exist within an individual's repertoire to be facilitated [16]. Thus, it is expected that the observation should be mapped on an individual's feature space to facilitate learning via reward estimation. Alissandrakis approached the correspondence problem by creating a corresponding library [17]. This is different from the proposed approach in which an attempt is made to determine a reward function as opposed to simply determining correspondence, i.e., relying on the generalization ability of forward RL. A study by Englert proposed directly matching robot trajectories with demonstrations [18]. The present study mainly focuses on differences in actuation, that is, on learning a robot controller from a learned probabilistic system model.

To the best of the authors' knowledge, previous studies of apprenticeship learning do not focus on the problem stated in the present study. However, similar arguments were presented in extant literature related to transfer learning [19]. For example, a study by Dai presented *translated learning* to transfer knowledge between different feature spaces [20]. The method is formulated as risk-minimization, and it demonstrates favorable results with respect to classification problem. Translated learning requires a paired sample (co-occurrence data) as is the case with the proposed method. However, there are transfer learning techniques that do not use paired samples [21]. A connection to these methods is promising in addressing the control problem of real robots.

Additionally, the proposed method can be combined with a state-of-the-art IRL based on distribution matching [22]. The proposed method relies on standard feature matching. Nevertheless, the study uses a probability distribution as a method to model expert behavior. Thus, it could be linked with the proposed approach that uses a conditional density estimation.

V. CONCLUSIONS

In this study, apprenticeship learning in distinct feature space is introduced in which an agent and an expert follow non-identical MDPs. In order to estimate the reward function in the scenario, a conditional density estimation technique is utilized to obtain feature expectation in an agent feature space. The feature expectation is represented in a closed-form. The proposed method is useful because it is simple and computationally efficient. Simulation results demonstrate that the proposed method enables the agent to infer reasonable policy from the expert although the MDP of the expert differs from that of the agent.

It is expected that the proposed method can be used for knowledge transfer among heterogeneous agents including robots and humans. Therefore, a future study will involve experiments on a real robot. It is expected that the proposed approach will contribute to improving the learning speed of forward RL by providing information that is useful for exploration purposes.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Number 16K16132.

REFERENCES

[1] M. Wiering, M. van Otterlo, Reinforcement learning: State of the art, Springer-Verlag, 2012.
[2] J. Peters, S. Schaal, Natural actor-critic, Journal of Neurocomputing, vol.71, no.7, pp.1180–1190, 2008.
[3] E. Theodorou, J. Buchli, S. Schaal, A generalized path integral control approach to reinforcement learning, Journal of Machine Learning Research, vol.11, pp.3137–3181, 2010.
[4] J. Kober, J.A. Bagnell, J. Peters, Reinforcement learning in robotics: A survey, The Int. Journal of Robotics Research, vol.32, no.11, pp.1238–1274, 2013.
[5] A. Ng, S. Russell, Algorithms for inverse reinforcement learning, Proc. of the 17th Int. Conf. on Machine Learning, pp.663–670, 2000.
[6] P. Abbeel, A. Ng, Apprenticeship learning via inverse reinforcement learning, Proc. of the 21st ACM Int. Conf. on Machine Learning, 2004.
[7] N. Ratliff, J. Bagnell, M. Zinkevich, Maximum margin planning, Proc. of the 23rd Int. Conf. on Machine Learning, pp.729–736, 2006.
[8] U. Syed, R.E. Schapire, A game-theoretic approach to apprenticeship learning, Advances in Neural Information Processing Systems 20, pp.1449–1456, 2008.
[9] B. Michini, J.P. How, Improving the efficiency of Bayesian inverse reinforcement learning, Proc. of the 29th IEEE Int. Conf. on Robotics and Automation, pp.3651–3656, 2012.
[10] M. Sugiyama, I. Takeuchi, T. Kanamori, T. Suzuki, H. Hachiya, D. Okanohara, Least-squares conditional density estimation, IEICE Trans. on Information and Systems, vol.E93-D, no.3, pp.583–594, 2010.
[11] R.C.L. Wolff, Q. Yao, P. Hall, Methods for estimating a conditional distribution function, Journal of the American Statistical Association, vol.94, no.445, pp.154–163, 1999.
[12] A. Boularias, J. Kober, J. Peters, Relative entropy inverse reinforcement learning, Proc. of the 14th Int. Conf. on Artificial Intelligence and Statistics, pp.182–189, 2011.
[13] D. Ramachandran, E. Amir, Bayesian inverse reinforcement learning, Proc. of the 20th Int. Joint Conf. on Artificial Intelligence, pp.2586–2591, 2007.
[14] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, M. Riedmiller, Deterministic policy gradient algorithms, International Conference on Machine Learning, 2014.
[15] M. Lopes, F. Melo, L. Montesano, Active learning for reward estimation in inverse reinforcement learning, Proc. of the 2009 European Conf. on Machine Learning, pp.31–46, 2009.
[16] R.W. Byrne, Imitation as behaviour parsing, Philosophical Transactions of the Royal Society of London B: Biological Sciences, vol.358, no.1431, pp.529–536, 2003.
[17] A. Alissandrakis, C.L. Nehaniv, K. Dautenhahn, Solving the correspondence problem in robotic imitation across embodiments: synchrony, perception and culture in artifacts, Imitation and Social Learning in Robots, Humans and Animals, chap. 12, pp.249–273, 2007.
[18] P. Englert, A. Paraschos, J. Peters, M.P. Deisenroth, Probabilistic model-based imitation learning, Adaptive Behavior, vol.21, no.5, pp.388–403, 2013.
[19] S. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. on Knowledge and Data Engineering, vol.22, no.10, pp.1345–1359, 2010.
[20] W. Dai, Y. Chen, G. Xue, Q. Yang, Y. Yu, Translated learning: Transfer learning across different feature spaces, Proc. of Advances in Neural Information Processing Systems 21, pp.353–360, 2009.
[21] K.D. Feuz, D.J. Cook, Transfer learning across feature-rich heterogeneous feature spaces via Feature-Space Remapping (FSR), ACM Trans. on Intelligent Systems and Technology, vol.6, no.1, pp.1–27, 2015.
[22] O. Arenz, H. Abdulsamad, G. Neumann, Optimal control and inverse optimal control by distribution matching, International Conference on Intelligent Robots and Systems, pp.4046–4053, 2016.