Extraction of Reward-Related Feature Space Using Correlation-Based and Reward-Based Learning Methods

Poramate Manoonpong[2,1], Florentin Wörgötter[1], and Jun Morimoto[2]

[1] Bernstein Center for Computational Neuroscience (BCCN),
III. Institute of Physics,
University of Göttingen, Göttingen 37077, Germany
{poramate,worgott}@physik3.gwdg.de

[2] ATR Computational Neuroscience Laboratories, 2-2-2 Hikaridai Seika-cho,
Soraku-gun, Kyoto 619-0288, Japan
xmorimo@atr.jp

Abstract. The purpose of this article is to present a novel learning paradigm that extracts reward-related low-dimensional state space by combining correlation-based learning like Input Correlation Learning (ICO learning) and reward-based learning like Reinforcement Learning (RL). Since ICO learning can quickly find a correlation between a state and an unwanted condition (e.g., failure), we use it to extract low-dimensional feature space in which we can find a failure avoidance policy. Then, the extracted feature space is used as a prior for RL. If we can extract proper feature space for a given task, a model of the policy can be simple and the policy can be easily improved. The performance of this learning paradigm is evaluated through simulation of a cart-pole system. As a result, we show that the proposed method can enhance the feature extraction process to find the proper feature space for a pole balancing policy. That is it allows a policy to effectively stabilize the pole in the largest domain of initial conditions compared to only using ICO learning or only using RL without any prior knowledge.

Keywords: Unsupervised learning, Reinforcement learning, Neural control, Sequential combination, Pole balancing.

1 Introduction

Living creatures, like humans and animals, can effectively learn solving a variety of tasks. They can learn a correlation between an earlier stimulus (called predictive signal) and a later one (called reflex signal) to react to the earlier stimulus, not having to wait for the later signal. For example, heat radiation (predictive signal) precedes a pain signal (reflex signal) when touching a hot surface. Thus, they learn an anticipatory action to avoid the late unwanted stimulus (i.e., avoiding to touch the hot surface). Such a learning mechanism is known as correlation-based learning (or temporal sequence learning). Furthermore, the

K.W. Wong et al. (Eds.): ICONIP 2010, Part I, LNCS 6443, pp. 414–421, 2010.
© Springer-Verlag Berlin Heidelberg 2010

living creatures also have the ability to learn to react appropriately to particular stimuli on the basis of associated rewards or punishments. This kind of learning strategy is minimally supervised since they are not explicitly taught. Instead, they must work this out for themselves on the basis of their past experiences (exploitation), new choices (exploration), and the reinforcement. This learning mechanism is known as reinforcement learning (RL). These two biological learning mechanisms have been applied to Artificial Intelligence (AI) from several points of view including the development of adaptive autonomous robots [1]. Most AI studies have *separately* used such learning mechanisms to allow robots to learn solving their tasks [2,3]. As a consequence, they might fail to solve some tasks required evaluative feedback when using only correlation-based learning. On the other hand, using RL without any prior knowledge (predefined control parameters) for high-dimensional continuous-state systems often requires long learning times. Thus a number of investigators have focused on building various low-level control parameters before applying RL [4,5].

In contrast to the robot learning strategy, living creatures probably combine both learning mechanisms in a way that fast correlation-based learning automates their intuition (i.e., providing low-level control parameters) which will guide RL for effectively solving complex tasks. Following this, we propose here how correlation-based learning (e.g., input correlation learning (ICO learning) [2]) and RL (e.g., actor-critic RL [6]) can be combined in a sequential way such that this learning paradigm extracts reward-related features to allow a policy to accomplish a given task. If we can extract proper feature space for a given task, a model of the policy can be simple, i.e., the policy with small number of parameters can be used to accomplish the given task. Advantage of using the simple policy is that it can be easily improved. We have chosen a pole balancing problem as a first test since balancing an inverted pendulum provides a well known class of control problems and often serves as a benchmark problem for dynamical control. However, the main purpose of this article is not to demonstrate the use of the combination between ICO learning and actor-critic RL for the pole balancing system but to suggest that this learning paradigm can be an efficient way to find reward-related feature space to solve dynamic sensorimotor control problems. Note that in this study, policy parameters are fixed and only the feature space is updated.

Before presenting the proposed learning strategy and its performance, in the following section we first show how ICO leaning can be applied and quickly learn to find the feature space for the pole balancing problem. Afterwards in section 3 we show how we sequentially combine ICO learning with actor-critic RL to modify the feature space to achieve better task performance. In section 4 we provide comparison results of different learning mechanisms, followed by conclusions.

2 Correlation-Based Learning to Extract Feature Space

Here we present how ICO learning can be applied to extract the reward-related feature space where its learning rule considers only cross-correlating two types of input signals with each other: earlier signals and a later one. As a concrete example, we consider the pole balancing problem [7] (see Fig. 1a). The task is to balance an inverted pendulum, which is mounted on a cart moving freely in a one-dimensional interval, and to simultaneously avoid the interval boundaries. The pole is free to move only in the vertical plane of the cart and track. This cart-pole system is simulated on a desktop PC with dual-core Intel processors at 2.4 GHz and updated by using Euler discretization with time steps of 0.01 s. The system provides four state variables: angle of the pole with the vertical (θ), pole angular velocity ($\dot{\theta}$), position of the cart on the track (x), and cart velocity ($\dot{x}$). The cart is bound to move in the interval $-2.4 < x < 2.4$ $[m]$ and the angle is allowed to vary in the interval $-12 < \theta < 12$ $[°]$. The simulated model includes all nonlinearities of the physical system (see [7] for the equations of this physical cart-pole system).

The feature extraction method based on ICO learning (see Fig.1a) for this dynamical system is modelled as a linear projection of four earlier signals (called predictive signals) which are the state variables (θ, $\dot{\theta}$, x, $\dot{x}$) to one-dimensional feature space. To update the feature space, we use a later signal (called a reward (penalty) signal, r) which is a signal given just before the system fails. The reward signal has a negative value (-1.0), if $x < -2.35$ m, $x > 2.35$ m, $\theta > 11.5°$, or $\theta < -11.5°$, and 0 otherwise. All the state inputs (θ, $\dot{\theta}$, x, $\dot{x}$) are scaled onto the interval $[-1, 1]$ as described in [8]. A projection from original state space to the low-dimensional feature space $\mathcal{Z}$ is specified by:

$$z(t) = \boldsymbol{w}^T \boldsymbol{x}(t), \tag{1}$$

where $z \in \mathcal{Z} \subset \mathcal{R}$, and $\boldsymbol{x}$ is the original state vector while $\boldsymbol{w}$ represents synaptic weights (projection vector). These weights which are initially set to 0.0 get changed by ICO learning using the cross-correlation between the predictive signals and a change of the reward signal. They are given by:

$$\boldsymbol{w}(t+1) = \boldsymbol{w}(t) + \mu|\boldsymbol{x}(t)|q, \tag{2}$$

$$q = |\min(0, \varDelta r(t))|, \tag{3}$$

where $\varDelta r(t) = r(t+1) - r(t)$ denotes the change of the reward signal and $\mu = 1.0 \times 10^{-4}$ is a learning rate.

Here we consider the reflex output $U(t)$ as a part of control output as suggested in [2]:

$$U(t) = \begin{cases} 1.0 & : \ x(t) \ < \ -2.35 \text{ m or } \theta(\text{t}) \ > \ 11.5° \\ -1.0 & : \ x(t) \ > \ 2.35 \text{ m or } \theta(\text{t}) \ < \ -11.5° \\ 0.0 & : \ \text{otherwise.} \end{cases} \tag{4}$$

By using domain knowledge, we can construct a failure avoidance policy of the cart-pole system in the low-dimensional feature space:

$$u(t) = Gz(t) + U(t), \tag{5}$$

where G is a parameter of the policy (i.e., gain in this case). Here, it is set to 10.0 [7]. According to this setup ICO learning will gradually develop the synaptic weights $\boldsymbol{w}$ (see Fig. 1) to obtain the appropriate projection vector from the original state space to the feature space for balancing the pole and also avoiding the cart to hit the interval boundaries (i.e., failure avoidance policy).

To test the performance of the proposed feature extraction method using ICO learning for this dynamical pole balancing task, we let it learn to balance the pole on 25 x 49 initial conditions (θ, x) represented by squares in Fig. 1b while $\dot{\theta}$ and $\dot{x}$

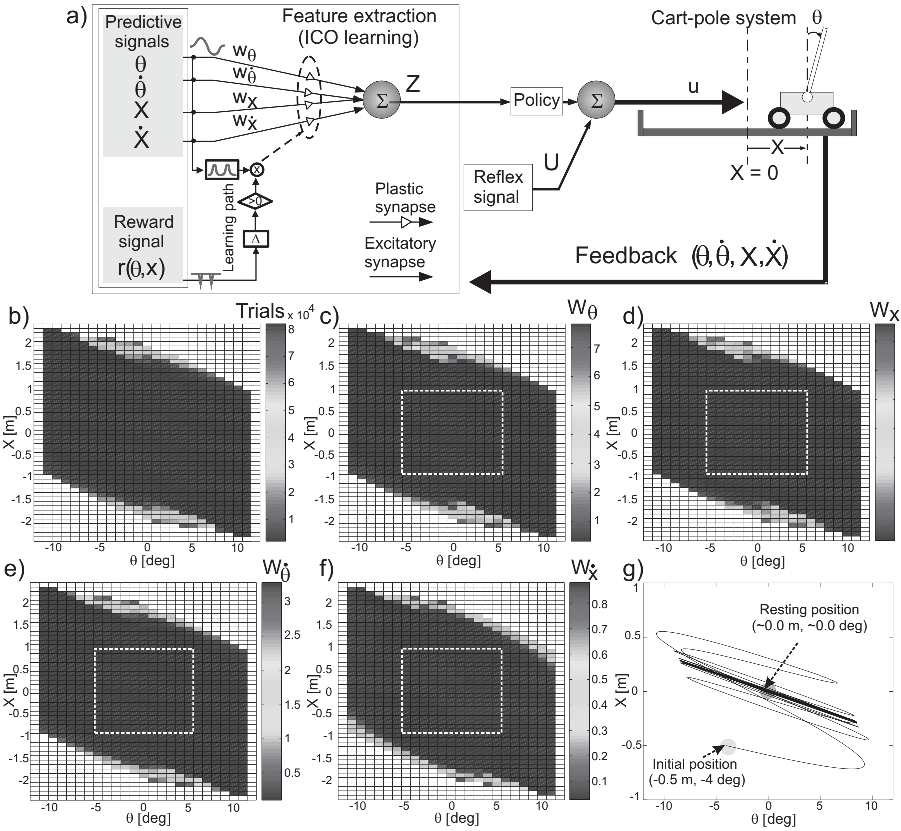

Fig. 1. (a) Feature extraction using ICO learning for the cart-pole system. Only one learning path instead of four is shown for clarity (see text for more details). (b) Performance of the policy on benchmark initial conditions. Colored area denotes a typical domain for successful control (≈ 68 %) where the color-coded bar presents the number of trials from start to success. White area represents a domain in which the policy fails to balance. (c)–(f) Resulting weights, i.e., projection vectors, $(w_\theta,\ w_x,\ w_{\dot{\theta}},\ w_{\dot{x}})$ after learning. Note that all weight values in the white lower left and upper right corners, where the system fails, are removed for clarity. The area inside a dashed frame of each diagram shows the weights which will serve to generate prior weight distribution for the reward-based learning. (g) Path in (x, θ)-space for arbitrary initial condition ($x = -0.5$ m, $\theta = -4$ deg) under control of the weight distribution.

are initially set to small random values representing the system noise. Each trial during a run starts with a given initial state and ends either in "success" (which occurs when the pole is kept in balance for at least 1000 seconds) or "failure" (which occurs when the pole falls 12 deg to either side or the cart moves 2.4 m to either side). A run at each initial condition is terminated when either a successful trial achieved or the maximum number of trials is reached (e.g., here 1.0×10^5 trials). During this learning process the system is reset to the same initial state at failure. We observe that the policy defined in the extracted feature space (see Eq. (5)) is able to balance the pole and avoid the ends of the interval in a relatively large (x, θ)–domain of initial conditions (see Fig. 1b). It is important to note that each trial generally uses so much less computing power resulting in a fast learning speed compared to other techniques (see comparison section below). Figures 1c–f show the resulting projection vector (or learned weights, w_θ, w_x, $w_{\dot{\theta}}$, $w_{\dot{x}}$) in the successful domain. Figure 1g exemplifies the behavior of the system displaying a path in (x, θ)-space for arbitrary initial condition (e.g., $x = -0.5$ m, $\theta = -4$ deg) of the cart-pole system.

3 Reward-Based Learning to Extract Feature Space

As shown in a previous section, one can see that the reward-related feature space extracted by ICO learning can be efficiently used for the pole balancing problem in a relatively large domain of initial conditions (see Fig. 1b). However, it still fails to stabilize the system at initial conditions in the critical corners of the benchmark domain (upper-right and lower-left areas in Fig. 1b). This is because correlation-based learning can only adapt the weights by recognizing a correlation between immediate reward (punishment) and it can not evaluate future (delayed) reward. Thus here we investigate whether the extracted feature space can be further modified so that the policy can stabilize the system in this domain.

To do so, we apply (continuous-state) reinforcement learning (RL) [6], [7] since its learning rule considering an association between stimuli and/or actions with the reinforcement that an agent receives can evaluate the future (delayed) reward. We use the actor-critic type RL that can be divided into two sub-mechanisms: the learning of the feature space (actor) and the learning of an evaluation function (critic). The feature extraction part is designed to have the same circuit as the feature extraction process of ICO learning (compare Figs. 1a and 2a).

For the critic network, we use a normalized Gaussian neural network (see Fig. 2a) as a function approximator to represent the value function or the prediction (V, see [6] for more details including equations). In this cart-pole system, the network has 162 hidden neurons ($H_{1,\ldots,162}$, see Fig. 2a) where centers are fixed on a grid according to the boxes approach [7]. The learning rate of this critic network is manually adjusted. It is set to, e.g., 0.6. The TD error is computed from the prediction as $\delta(t) = r(t) + \gamma V(t) - V(t-1)$. r is an external reinforcement signal (-1 when failure occurs, 0 otherwise) [7]. γ is a discount factor, i.e., distant rewards are less important. We set it to 0.95 based on [7].

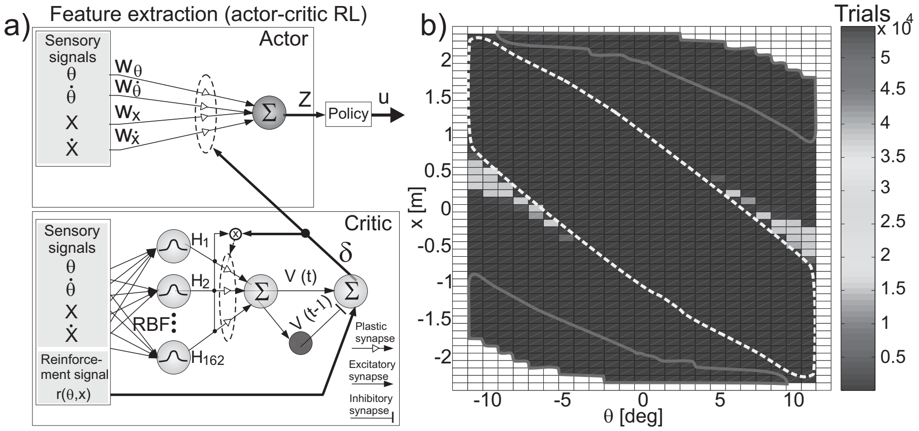

Fig. 2. (a) Feature extraction using actor-critic RL for the cart-pole system. (b) Performance of the policy defined in the extracted feature space. The prior weight distribution are given by the learned weights by ICO learning. Colored area denotes a typical domain for successful control (≈ 84%) where the color-coded bar presents the number of trials until success. White area represents the domain in which the policy fails to balance. Area inside a white dashed frame shows the initial conditions on which the policy can stabilize the system without actor-critic RL. Area inside red frames shows improvement of the feature space achieved by actor-critic RL.

Here we modify the feature space $\mathcal{Z}$ through Bayesian update given by:

$$P(\boldsymbol{w}|u^{ob},\boldsymbol{x}) = \frac{P(u^{ob}|\boldsymbol{w},\boldsymbol{x})P(\boldsymbol{w})}{\int P(u^{ob}|\boldsymbol{w},\boldsymbol{x})P(\boldsymbol{w})d\boldsymbol{w}}, \tag{6}$$

where $P(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_{\boldsymbol{w}}, \Sigma_{\boldsymbol{w}})$ is the prior probability of weight distribution given by ICO learning for the first iteration. Since ICO learning tends to acquire different weighs when the learning process starts from different initial conditions, we estimate mean $\boldsymbol{\mu}_{\boldsymbol{w}}$ and variance $\Sigma_{\boldsymbol{w}}$ from the learned weight vectors. $P(u|\boldsymbol{w},\boldsymbol{x}) = \mathcal{N}(u|\mu_u, \sigma_u)$ is the control policy. Note that we use the result of ICO learning as the prior only for the first update. From the second update, we use the weight distribution derived in previous iteration as the prior.

Here we consider the feature $z = \boldsymbol{w}^T\boldsymbol{x}$. The mean output of the policy is designed as $\mu_u = Gz$. The observation of the control output u^{ob} in Eq. (6) is acquired from the RL framework:

$$u^{ob} = \mu_u + \Delta u, \tag{7}$$

where

$$\Delta u = \alpha\delta(u - \mu_u). \tag{8}$$

Here δ is the TD error and α is a scaling parameter that corresponds to the learning rate.

Figure 2b shows the performance of the policy defined in the feature space using actor-critic RL where the prior weight distribution $P(\boldsymbol{w})$ are derived from ICO learning. It can be seen that the policy can now stabilize the system in a larger domain ($\approx$ 84%, i.e., $\approx$ 16% more, see red frames in Fig. 2b) including some parts of the critical initial conditions. The remaining parts (white areas) seems to be difficult to achieve by using linear control. The results we obtained here are comparable to [8] where linear control using an evolutionary algorithm for weight adaptation is employed. Furthermore, we observe that actor-critic RL only starts to optimize the weights in the domain where the initial feature space is not proper to stabilize the system (colored area outside a white dashed frame in Fig. 2b). In fact there are a few initial conditions of the system where actor-critic RL requires a lot of trials ($>$ 2000 trials, see triangular areas near the dashed frame in Fig. 2b) while most of them can be achieved after around 5–2000 trials. By contrast, if the actor weights are not appropriately given at the beginning (e.g., initially setting them to 0.0) actor-critic RL needs to learn in the whole domain. It also requires much more trials for each given initial condition and the policy can stabilize the system in a smaller domain (see Fig. 3). From this point of view, our experimental results suggest that providing the appropriate prior to actor-critic RL can speed up the learning process and also allows it to efficiently extract feature space.

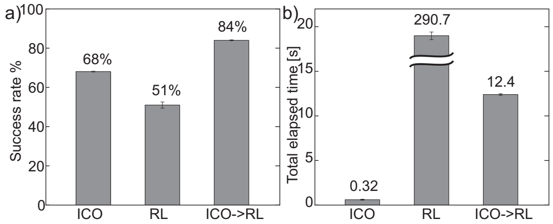

Fig. 3. (a) Histogram showing the average of success rate of each learning model; i.e., percentage of success in the total 25 x 49 set of initial conditions ((x, θ)–domain). (b) Histogram showing the average of the total elapsed simulated physical time of all success for each learning model. The elapsed times are recorded from starting until ending in success where failure cases are ignored. *ICO* and *RL*: the feature extraction using ICO learning and actor-critic RL, respectively, where all weights are initially set to 0.0, *ICO* $\rightarrow$ *RL*: the feature extraction using actor-critic RL but all weights are predefined by the weight distribution obtained from ICO learning. Note that in this comparison, all learning models use the same parameters, like learning rate and discount factor.

4 Comparison Results

In this section, we compare the performance of this learning model with the original ones. The results are shown in Fig. 3. It can be seen that ICO learning can quickly learn to find appropriate feature space in a relatively large domain of initial conditions while actor-critic RL is very slow and can achieve success only in a smaller domain if we limit the maximum number of trials. However, the performance of the policy can be strongly improved by using prior weight distribution generated by ICO learning for actor-critic RL. As a consequence, the policy succeeds for the larger initial condition domain.

5 Conclusions

In this study, we proposed a new learning paradigm that sequentially combines ICO learning and actor-critic RL to extract feature space for a dynamical system. In concrete, we consider the pole balancing task as the dynamical system. To a certain extent the experimental studies pursued here sharpen our understanding of how correlation-based learning can be combined with RL to find the low-dimensional feature space. In future work, we will investigate the theoretical properties of this learning model and its dynamical behavior. We will also apply this learning strategy to real robotic tasks, like adaptive walking or mobile robot control.

Acknowledgments. This research was supported by Japan Society for the Promotion of Science (JSPS), the EU PACO-PLUS project, Bernstein Center for Computational Neuroscience II Göttingen (BCCN, project D1), the German Research Foundation (DFG, SE1042/05) and the Strategic Research Program for Brain Sciences (SRPBS).

References

1. Pfeifer, R., Lungarella, M., Iida, F.: Self-Organization, Embodiment, and Biologically Inspired Robotics. Science 318(5853), 1088–1093 (2007)
2. Porr, B., Wörgötter, F.: Strongly Improved Stability and Faster Convergence of Temporal Sequence Learning by Using Input Correlations Only. Neural Comput. 18(6), 1380–1412 (2006)
3. Phon-Amnuaisuk, S.: Learning Cooperative Behaviours in Multiagent Reinforcement Learning. In: Leung, C.S., Lee, M., Chan, J.H. (eds.) ICONIP 2009. LNCS, vol. 5863, pp. 570–579. Springer, Heidelberg (2009)
4. Kolter, J.Z., Ng, A.Y.: Policy Search via the Signed Derivative. In: The Proc. of Robotics: Science and Systems (RSS), p. 27 (2009)
5. Melo, F.S., Lopes, M., Santos-Victor, J., Ribeiro, M.I.: A Unified Framework for Imitation-Like Behaviours. In: The Proc. of 4th International Symposium on Imitation in Animals and Artifacts, pp. 241–250 (2007)
6. Doya, K.: Reinforcement Learning in Continuous Time and Space. Neural Comput. 12, 219–245 (2000)
7. Barto, A.G., Sutton, R.S., Anderson, C.: Neuron-Like Adaptive Elements That Can Solve Difficult Learning Control Problems. IEEE Transactions on Systems, Man, and Cybernetics 13, 834–846 (1983)
8. Pasemann, F.: Evolving Neuropolicys for Balancing an Inverted Pendulum. Network: Computation in Neural Systems 9, 495–511 (1998)