Reinforcement Learning in Fine Time Discretization

Paweł Wawrzyński

Institute of Control and Computation Engineering,
Warsaw University of Technology,
Nowowiejska 15/19, 00-665 Warsaw, Poland
p.wawrzynski@elka.pw.edu.pl

Abstract. Reinforcement Learning (RL) is analyzed here as a tool for control system optimization. State and action spaces are assumed to be continuous. Time is assumed to be discrete, yet the discretization may be arbitrarily fine. It is shown here that stationary policies, applied by most RL methods, are improper in control applications, since for fine time discretization they can not assure bounded variance of policy gradient estimators. As a remedy to that difficulty, we propose the use of piecewise non-Markov policies. Policies of this type can be optimized by means of most RL algorithms, namely those based on likelihood ratio.

1 Introduction

Reinforcement Learning (RL) algorithms provide solutions to the problem of an intelligent agent that optimizes its behavior in an initially unknown environment. Adaptive control is a very important application of the intelligent agent problem. We would like to construct controllers that are able to "learn" by trial and error to control plants whose dynamics is unknown. The controller may be understood as the agent, and it is rewarded for reaching the control objectives. In present control applications, with fast digital controllers, control stimuli are applied with high frequency. Therefore, each agent's state results from thousands of previous actions rather that tens like in board games often analyzed as benchmark problems in RL.

Most RL algorithms [1,9,4,5,7] optimize stationary policies, i.e. ones that draw an action only on the basis of a current state. Application of RL in control systems requires discretization of time. Good control requires fine time discretization. However, a stationary stochastic policy applied to a deterministic system leads to a deterministic behavior of the system for diminishing time discretization [6]. Clearly, this phenomenon precludes exploration capabilities of such policies. Here we analyze the influence of time discretization on variance of policy gradient estimators. In an example we show that stabilization of this variance quickly becomes infeasible as the time discretization decreases.

Our remedy for the fine time discretization problem is based on defining a policy in a special way. Namely, the policy divides agent–environment interaction

B. Beliczynski et al. (Eds.): ICANNGA 2007, Part I, LNCS 4431, pp. 470–479, 2007.

into periods such that it relates actions with each others within the same period. Within each period a coherent experiment is carried out that gives a clue to policy improvement. On the basis of a given Markov Decision Process (MDP) and such the policy we define a new MDP and a stationary policy in the new one. We show that each RL algorithm based on likelihood ratio can be applied to optimize the stationary policy in the new MDP.

2 Problem Statement and Likelihood Ratio

We will consider the standard episodic RL setup [8]. A Markov Decision Process is a tuple $\langle \mathcal{S}, \mathcal{A}, P_s, r, P_0, \mathcal{S}^* \rangle$ where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, respectively; $\{P_s(\cdot|s,a) : s \in \mathcal{S}, a \in \mathcal{A}\}$ is a set of state transition probabilities; we write $s_{t+1} \sim P_s(\cdot|s_t, a_t)$. In this work we assume that both $\mathcal{S}$ and $\mathcal{A}$ are multidimensional continuous and each P_s is a density. The immediate reward, r_t depends on the action and the next state, $r_t = r(a_t, s_{t+1})$. P_0 is the distribution of first states of each episode and $\mathcal{S}^*$ is the set of terminal states. The objective of a reinforcement learning is to find a control policy that maximizes future rewards in each state.

We are interested in applications of the solution of the above RL problem to learning control tasks. A painful difficulty that emerges in control problems is the fine time discretization. It makes a single action impact the overall performance insignificantly. Furthermore, the impact of the action emerges a large number (thousands) of steps after the very action took place. We require the learning algorithm work properly no matter how fine the time discretization is.

The problem of Reinforcement Learning is an issue of optimization of a certain performance measure with respect to policy parameters. Because the probability distributions that define the RL problem at hand are unknown, the optimization can not be done directly. A possible approach is to adjust policy parameters along gradient estimators of the performance measure. An important class of such estimators is based on, so called, *likelihood ratio*. Let $f(a;\theta)$ be a density of random variables a of values in $\mathcal{A}$. f is parametrized by vector $\theta \in \mathfrak{R}^{n_\theta}$. A sample, a, yields a payment, $r(a)$. We are interested in maximization of the expected payment

$$J(\theta) = \mathcal{E}_\theta r(a) = \int\limits_{\mathcal{A}} r(a) f(a;\theta)\,\mathrm{d}a.$$

Under certain, quite liberal regularity conditions, for each constant c,

$$\nabla J(\theta) = \nabla(J(\theta) - c) = \int\limits_{\mathcal{A}} (r(a) - c) \nabla_\theta f(a;\theta)\,\mathrm{d}a = \int\limits_{\mathcal{A}} \left((r(a) - c) \frac{\nabla_\theta f(a;\theta)}{f(a;\theta)} \right) f(a;\theta)\,\mathrm{d}a$$

and thus

$$(r(a) - c) \frac{\nabla_\theta f(a;\theta)}{f(a;\theta)} = (r(a) - c) \nabla_\theta \ln f(a;\theta)$$

is an unbiased estimator of the gradient $\nabla J(\theta)$. Its variance might be minimized by an appropriate choice of c. The term $\nabla_\theta \ln f(a;\theta)$ is the likelihood ratio.

Reference [2] contains an interesting discussion about the history of its use in RL and other fields.

3 A Stationary Policy for a Continuous-Time System

In this section we analyze by means of a simple example, how the time discretization influences policy gradient estimation. An important insight to this issue has been provided in [6] where it has been shown that in a continuous environment, under quite general conditions, the state trajectory converges to a deterministic limit as the time discretization diminishes. The question arise how this phenomenon influences quality of policy gradient estimators. A general answer to this question is difficult to provide. However, the simple example below suggests that this influence can be demaging.

Let state represent one-dimensional velocity and action represent one-dimensional acceleration. We have $\mathcal{S} = \mathcal{A} = \Re$. An episode lasts for 1 sec. and it includes T steps, $\delta = 1/T$ long each. Within each step an action is drawn from the normal distribution $N(\theta, \sigma_a^2)$ where θ is a policy parameter. The action defines constant acceleration within a step and velocity at the beginning of a trial is null. The only nonzero reward is equal to noised velocity in the last state. We have

$$s_t = \begin{cases} 0 & \text{for } t = 0 \\ s_{t-1} + \delta a_t & \text{for } t > 0, \end{cases} \qquad r_t = \begin{cases} 0 & \text{for } t < T-1 \\ s_T + y_T & \text{for } t = T-1 \end{cases}$$

where y is a random variable drawn from the normal distribution $N(0, \sigma_y^2)$.

The quality index, $J(\theta)$, of the policy defined by θ is equal to the expected reward at the end of an episode. Because the final reward is a sum of random variables, we have

$$\mathcal{E}_\theta r_{T-1} = \mathcal{E}_\theta \left(\sum_{t=0}^{T-1} \delta a_t + y_T \right) = T\delta\theta = \theta.$$

Therefore, $J(\theta) = \theta$ and $\nabla J(\theta) = 1$. Our main concern here is variance of the policy gradient estimator. We will consider the policy gradient estimator applied in the REINFORCE algorithm [10] since prevailing policy gradient estimators can be considered modifications of this early formula. The estimator applied to our problem is of the form

$$\widehat{g} = (r_{T-1} - c) \sum_{t=0}^{T-1} \frac{\partial \ln \pi(a_t; s_t, \theta)}{\partial \theta^T}$$

where c is the *baseline*. Variance of this estimator is defined by the following formula

$$\mathcal{V}\widehat{g} = 2 + \frac{1}{\delta\sigma_a^2} \left((c-\theta)^2 + \sigma_y^2 \right) \tag{1}$$

derived in the Appendix. We can see that the larger action variance, the smaller gradient estimator variance. The exploration–exploitation balance becomes conspicuous when we compare $\mathcal{V}\widehat{g}$ with variance of s_T, namely $\mathcal{V}s_T = \delta\sigma_a^2$ (see the Appendix). By comparing this value with (1), we can see that $\mathcal{V}s_T$ is "almost" inversely proportional to $\mathcal{V}\widehat{g}$.

What happens when the time discretization parameter δ decreases? In order to keep gradient estimator variance small, variance of action has to be increased. In fact, variance of gradient estimator remains constant if only

$$\sigma_a^2 \propto 1/\delta.$$

Interestingly enough, this way variance of s_T is also stabilized.

It is seen in our example that in order to keep variance of policy gradient estimator bounded, we have to increase variance of action. However, "actions" in control systems are always bounded; hence, they can not have too large variance either. Therefore, while fine time discretization is necessary for good control it contradicts with quality of policy gradient estimation.

4 MDP Defined by Non-Markov Periods

Let the policy applied by the agent be *piecewise non-Markov* in the following sense. It divides an episode into periods and generates actions within each period on the basis of previous actions and states in this period. Let the periods be indexed by k and k-th period starts at time t_k and lasts for l_k instants. For $i : 0 \leq i < l_k$ we have[1]

$$a_{t_k+i} \sim \pi(\cdot\,; s_{t_k}, a_{t_k}, \ldots, s_{t_k+i}, \theta).$$

Let the periods defined by a piecewise non-Markov policy be called *non-Markov periods* or, in short, *nm-periods*.

Given a Markov Decision Process $M = \langle \mathcal{S}, \mathcal{A}, P_s, r, P_0, \mathcal{S}^* \rangle$ we define a new one, $\bar{M} = \langle \bar{\mathcal{S}}, \bar{\mathcal{A}}, \bar{P}_s, \bar{r}, P_0, \mathcal{S}^* \rangle$ with the use of nm-periods defined above. Let states and actions in $\bar{M}$ be denoted by $\bar{s}$ and $\bar{a}$, respectively, and time be indexed by k. Simultaneously, given a piecewise non-Markov policy π in M, we define a stationary policy $\bar{\pi}$ in $\bar{M}$ generating actions from $\bar{\mathcal{A}}$. States in $\bar{M}$ corresponds to first states in nm-periods; we have

$$\bar{s}_k = s_{t_k} \quad \text{and} \quad \bar{\mathcal{S}} = \mathcal{S}.$$

Actions in $\bar{M}$ corresponds to joint trajectories of states and actions within nm-periods, namely

$$\bar{a}_k = \langle a_{t_k}, s_{t_k+1}, \ldots, a_{t_k+l_k-1} \rangle \quad \text{and} \quad \bar{\mathcal{A}} = \bigcup_{i \geq 0} \mathcal{A} \times (\mathcal{S} \times \mathcal{A})^i.$$

[1] We apply "..." also to denote a subsequence of the sequence $(s_1, a_1, s_2, a_2, \ldots)$.

The transition distribution in $\bar{M}$, $\bar{P}_s$ is defined by P_s, π, and the way l_k emerges. In the simplest case $l_k = l$ for a certain constant l unless k-th period is the last one in the episode; then $1 \leq l_k \leq l$. We are free to define the method of calculating rewards in $\bar{M}$. For instance, a reward in $\bar{M}$ can be an average value of rewards gathered within the corresponding nm-period in M. The distribution of first states P_0 and the set of terminal states $\mathcal{S}^*$ remain unchanged.

An action in $\bar{M}$ is generated by the policy π in tandem with P_s. What is yet important is that we can calculate the likelihood ratio $\nabla_\theta \ln \bar{\pi}(\bar{a}_k; \bar{s}_k, \theta)$. Let us denote

$$S_k = [s_{t_k}^T, \ldots, s_{t_k+l_k-1}^T]^T, \quad A_k = [a_{t_k}^T, \ldots, a_{t_k+l_k-1}^T]^T, \tag{2}$$

$$\pi_A(A_k; S_k, \theta) = \prod_{i=0}^{l_k-1} \pi(a_{t_k+i}; s_{t_k}, a_{t_k}, \ldots, s_{t_k+i}, \theta). \tag{3}$$

π_A is a density of a sequence of actions within an nm-period given a sequence of states. It is entirely defined by the way the policy π generates actions. From the decomposition

$$\bar{\pi}(\bar{a}_k; \bar{s}_k, \theta) = \prod_{i=0}^{l_k-1} \pi(a_{t_k+i}; s_{t_k}, a_{t_k}, \ldots, s_{t_k+i}\theta) \prod_{i=0}^{l_k-2} P_s(s_{t+i+1}|s_{t_k+i}, a_{t_k+i}) \tag{4}$$

we see that

$$\nabla_\theta \ln \bar{\pi}(\bar{a}_k; \bar{s}_k, \theta) = \nabla_\theta \ln \pi_A(A_k; S_k, \theta).$$

A special feature of $\bar{M}$ is the fact that the agent is not entirely free to choose an action from $\bar{\mathcal{A}}$. It is hence impossible to apply *Q-Learning* [9] or SARSA to $\bar{M}$. However, optimization of a stationary policy in $\bar{M}$ can be in principle performed by all methods based on likelihood ratio, including episodic REINFORCE [10], Actor-Critics [4,5,7], OLPOMDP [3] and others.

5 Piecewise Non-Markov Policies – Examples

In the present section we define a simple class of non-Markov policies. The policies we suggest exploit each k-th period to carry out a coherent experiment that provides a clue to an improvement of the policy. This coherence is a consequence of the fact that while at each moment the action has a random component, there is a stochastic dependence among these components within the same nm-period. In the next subsection we analyze a way of generating such the stochastically dependent components.

Piecewise independent autoregressive process. Let $\{\epsilon_t, t = 1, 2, \ldots\}$ be a sequence of independent random vectors in $\mathfrak{R}^n$ drawn from the normal distribution with zero mean and covariance matrix Σ, i.e. $N(0, \Sigma)$. Also, let $\alpha \in (0, 1)$ and $\{\xi_t, t = 1, 2, \ldots\}$ be a sequence of random vectors in $\mathfrak{R}^n$ computed as

$$\xi_t = \begin{cases} \epsilon_t & \text{if } t{=}t_k \text{ for any } k \\ \alpha\xi_{t-1} + \sqrt{1-\alpha^2}\epsilon_t & \text{otherwise.} \end{cases} \tag{5}$$

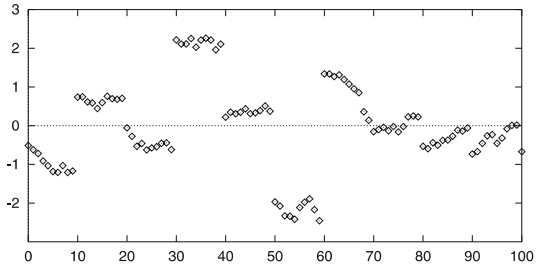

Fig. 1. A run of a piecewise independent autoregressive process for $\Sigma = 1, \alpha = 0.99, t_{k+1} - t_k \equiv 10$. Within an nm-period there is a correlation between random elements while there is no correlation between elements in different nm-periods.

From the above definition it is easy to see, that $\mathcal{E}\xi_t = 0$ for all t. Also, each ξ_t only depends on ϵ-s that belong to the same nm-period. Therefore,

$$\mathrm{cov}(\xi_t, \xi_{t'}) = \mathcal{E}(\xi_t \xi_{t'}^T) = 0$$

for t and t' in different nm-periods. Let us find $\mathrm{cov}(\xi_t, \xi_{t'})$ for t, t' in the same, k-th nm-period. We have

$$\begin{aligned}\xi_t &= \alpha\xi_{t-1} + \sqrt{1-\alpha^2}\epsilon_t = \alpha^2\xi_{t-2} + \alpha\sqrt{1-\alpha^2}\epsilon_{t-1} + \sqrt{1-\alpha^2}\epsilon_t \\ &= \cdots = \alpha^{t-t_k}\epsilon_{t_k} + \sqrt{1-\alpha^2}\sum_{i=0}^{t-t_k-1}\alpha^i\epsilon_{t-i}.\end{aligned}$$

If $t \le t'$, then

$$\mathrm{cov}(\xi_t, \xi_{t'}) = \mathcal{E}(\epsilon\epsilon^\top)\left(\alpha^{t+t'-2t_k} + (1-\alpha^2)\sum_{i=0}^{t-t_k-1}\alpha^{2i+t'-t}\right) = \Sigma\alpha^{t'-t}.$$

The result for $t' \le t$ is symmetrical. Generally, for t, t' in the same nm-period,

$$\mathrm{cov}(\xi_t, \xi_{t'}) = \alpha^{|t'-t|}\Sigma \tag{6}$$

and thus $\{\xi_t, t = t_k, \ldots, t_{k+1} - 1\}$ happens to be an autoregressive stochastic process. Notice that $\mathrm{cov}(\xi_t, \xi_t) \equiv \Sigma \equiv \mathrm{cov}(\epsilon_t, \epsilon_t)$.

The random process we defined above, the piecewise independent autoregressive process, may be interpreted as a simple method of transforming normal white noise, ϵ_t, into sequences of random vectors that are stochastically dependent (see Fig. 1). Thank to that dependence, each of these sequences may support a coherent experiment that gives a clue to policy improvement. Below we present two policy that make use of such experiments.

Deterministic Transformation + Stochastic Process. Let an action, a_t, be calculated as

$$a_t = \widetilde{a}(s_t; \theta) + \xi_t \tag{7}$$

where $\widetilde{a} : \mathcal{S} \times \Theta \mapsto \mathcal{A}$ is a certain deterministic function, e.g. a neural network with input s and weights θ. We will denote by $\nabla\widetilde{a}(s,\theta)$ a matrix of derivatives of $\widetilde{a}$ with respect to its second argument, namely

$$\nabla\widetilde{a}(s,\theta) = \frac{\partial\widetilde{a}(s,\theta)}{\partial\theta^T} = \left[\frac{\partial\widetilde{a}_j(s,\theta)}{\partial\theta_i}\right]_{i,j}.$$

What we need is to define the distribution $\pi_A(A_k; S_k, \theta)$ and the likelihood ratio $\nabla_\theta \ln \pi_A(A_k; S_k, \theta)$. for S_k and A_k defined in (2). Given trajectory S_k, quantities a_t result from adding random elements ξ_t to constant values $\widetilde{a}(s_t;\theta)$. Consequently, the distribution $\pi_A(A_k; S_k, \theta)$ is the normal one with the expected value and the covariance matrix equal to

$$\mathcal{E}A_k = m_k(\theta) = \begin{bmatrix} \widetilde{a}(s_{t_k},\theta) \\ \vdots \\ \widetilde{a}(s_{t_k+l_k-1},\theta) \end{bmatrix}, \quad \operatorname{cov} A_k = C_k = \begin{bmatrix} \Sigma & \alpha\Sigma & \cdots & \alpha^{l_k-1}\Sigma \\ \alpha\Sigma & \Sigma & & \\ \vdots & & \ddots & \vdots \\ \alpha^{l_k-1}\Sigma & & \cdots & \Sigma \end{bmatrix},$$

respectively. $\operatorname{cov} A_k$ results from the fact that $\operatorname{cov}(a_t, a_{t'}) = \operatorname{cov}(\xi_t, \xi_{t'})$ and (6).

We thus deal with the normal distribution $N(\mathcal{E}A_k, \operatorname{cov} A_k)$ whose mean depends on the parameter θ and variance does not. The density of this distribution is given by

$$\pi_A(A;S_k,\theta) = \left(\sqrt{2\pi}^{l_k n}|C_k|\right)^{-1} \exp\left(-0.5(A-m_k(\theta))^T (C_k)^{-1}(A-m_k(\theta))\right). \tag{8}$$

We also have

$$\begin{aligned} \nabla_\theta \ln \pi_A(A; S_k, \theta) &= (\nabla_\theta m_k(\theta))\, C_k^{-1}(A - m_k(\theta)) \\ &= [\nabla\widetilde{a}(s_{t_k};\theta)\cdots\nabla\widetilde{a}(s_{t_k+l_k-1};\theta)]C_k^{-1}(A-m_k(\theta)). \end{aligned} \tag{9}$$

It seems the most convenient to compute vector $(C_k)^{-1}(A-m_k(\theta))$ as y satisfying the linear equation

$$C_k y = A - m_k(\theta).$$

Deterministic Transformation Of Stochastic Process. Let an action, a_t, be calculated as

$$a_t = \widetilde{a}(s_t; \theta + \xi_t). \tag{10}$$

Here, the function $\widetilde{a} : \mathcal{S}\times\Theta \mapsto \mathcal{A}$ is defined as previously. However, what is noised here is parameters of $\widetilde{a}$ rather than its output. Therefore, the dimension of ξ_t is different than in the previous section. Here $\dim \xi_t = \dim \Theta$ while previously $\dim \xi_t = \dim \mathcal{A}$.

For $\widetilde{a}$ smooth with respect to its second argument it is true that

$$\widetilde{a}(s_t;\theta+\xi_t) \cong \widetilde{a}(s_t;\theta) + \nabla\widetilde{a}(s_t,\theta)\xi_t. \tag{11}$$

We will derive $\pi_A(A_k; S_k, \theta)$ and $\nabla_\theta \pi_A(A_k; S_k, \theta)$ assuming that the approximate equation (11) is strict. This assumption is satisfied for $\widetilde{a}$ linear in θ. Actions a_t result from an affine transformation of the normal elements ξ_t. Given S_k, the distribution of a_t is also normal with means and covariances equal to

$$\begin{aligned} \mathcal{E}a_t &\cong \widetilde{a}(s_t, \theta) + \nabla\widetilde{a}(s_t, \theta)\mathcal{E}\xi_t = \widetilde{a}(s_t, \theta) \\ \operatorname{cov}(a_t, a_{t'}) &\cong \nabla\widetilde{a}(s_t, \theta)\mathcal{E}\xi_t\xi_{t'}^T\nabla\widetilde{a}(s_{t'}, \theta)^T = \nabla\widetilde{a}(s_t, \theta)\alpha^{|t-t'|}\Sigma\nabla\widetilde{a}(s_{t'}; \theta)^T \end{aligned}$$

respectively, for t and t' in the same nm-period. If, additionally, $\Sigma = I\sigma^2$, covariance $\operatorname{cov}(a_t, a_{t'})$ is equal to

$$\operatorname{cov}(a_t, a_{t'}) = \sigma^2\alpha^{|t-t'|}\nabla\widetilde{a}(s_t, \theta)\nabla\widetilde{a}(s_{t'}, \theta)^T.$$

The above equation is important because it allows us to avoid computations with Σ which may be a large matrix.

Because A_k is a concatenation of a_t for a sequence of t, the distribution $\pi_A(A_k; S_k, \theta)$ is normal with mean and variance equal to

$$\mathcal{E}A_k = m_k(\theta) = \begin{bmatrix} \widetilde{a}(s_{t_k}, \theta) \\ \vdots \\ \widetilde{a}(s_{t'_k}, \theta) \end{bmatrix}, \quad \operatorname{cov}A_k = C_k = \begin{bmatrix} \operatorname{cov}(a_{t_k}, a_{t_k}) & \cdots & \operatorname{cov}(a_{t'_k}, a_{t_k}) \\ \vdots & \ddots & \vdots \\ \operatorname{cov}(a_{t_k}, a_{t'_k}) & \cdots & \operatorname{cov}(a_{t'_k}, a_{t'_k}) \end{bmatrix}$$

respectively, where $t'_k = t_k + l_k - 1$. With the above definition of $m_k(\theta)$ and C_k, the density $\pi_A(A; S_k, \theta)$ and the gradient $\nabla_\theta \ln \pi_A(A; S_k, \theta)$ are expressed in (8) and (9), respectively.

6 Discussion

Within the idea presented in this paper a reinforcement learning problem at hand is transformed into the other one and solved by one of the methods based on likelihood ratio. The objective of this transformation is to make RL algorithms better suited to adaptive control problems.

Piecewise non-Markov policies decrease the threat of deterministic behavior of the overall agent-environment system for fine time discretization. For instance, if nm-periods last for $\Delta\tau$ of real time regardless the discretization and there is a strong stochastic dependence between actions within the periods, the thread of deterministicity is headed off and quality of policy gradient estimators is preserved.

The notion of nm-periods introduces a certain additional degree of freedom into a RL process. Each such process includes a sequence of experiments that give clues to policy improvements, usually in the form of policy gradients. Let us consider the question: What should be the length of each such experiment? The answer given by the basic form of Episodic REINFORCE is: the length of an entire episode. Almost all the rest of RL algorithms give the answer that the experiment should last exactly one time step. Within the proposed approach the

answer is between these two extreme possibilities: An experiment lasts for l_k instants where l_k is a controllable parameter.

Third, dividing time into nm-periods enables more flexible treatment the concepts of time and reward. Let us consider algorithms operating on discounted rewards like OLPOMDP or Actor-Critics. The discount factor applied there defines how long, in terms of time, the agent looks ahead optimizing its actions. But it is often natural to look ahead in terms of space rather than time. At present instant t it may be less important what will happen when state becomes far from s_t while it may be quite soon in terms of time. In order to achieve the effect of looking ahead in terms of space, we can define length of a nm-period as

$$l_k = \min\{l : d(s_{t_k}, s_{t_k+l}) > \epsilon\} \tag{12}$$

where d is a certain metric in $\mathcal{S}$ and $\epsilon > 0$ is a threshold. Then, time goes by as fast as state changes. Furthermore, suppose we want to penalize the agent for growing time of accomplishing a certain task. It is possible by assigning a constant penalty to each moment the task is not completed. Apparently, the overall penalty is then proportional to the time of accomplishing the task. But what if we want this penalty to be in a different way related to this time? Within the traditional approach it would be quite problematic. Within our approach, we may define l_k as (12) and introduce the penalty as any function of l_k.

At present we carry out experiments with the proposed methodology. We test it with the use of a simulated 6-degree-of-freedom robotic manipulator. It appears that the performance of existing RL methods in optimization of a stationary control policy for an object of this kind is disappointing. However, when the concept of non-Markov periods is applied, the same methods become surprisingly efficient. We are going to report this work in another paper.

7 Conclusions

We have shown that in RL issues with fine time discretization, keeping variance of policy gradient estimator small may require unfeasibly large action variance. Significance of this difficulty comes from the fact that fine time discretization is typical in control problems. We have proposed a remedy, namely piecewise non-Markov policies. We have shown that combination of existing RL methods with the piecewise non-Markov policies introduces a new degree of freedom into a learning process, namely a length of a sequence of actions that give a clue to policy improvement. Therefore, the piecewise non-Markov policies may be treated on their own right as an enhancement of the existing RL methods.

References

1. A. G. Barto, R. S. Sutton, and C. W. Anderson. Neuronlike Adaptive Elements That Can Learn Difficult Learning Control Problems. *IEEE Transactions on System Man, and Cybernetics,* vol. SMC-13:834-846, 1983.

2. J. Baxter and P. L. Bartlett. Infinite-Horizon Policy-Gradient Estimation, *Journal of Artificial Intelligence Research,* vol. 15:319-350, 2001.
3. J. Baxter, P. L. Bartlett, & L. Weaver. Experiments with Infinite-Horizon, Policy-Gradient Estimation, *Journal of Artificial Intelligence Research,* vol. 15:351-381, 2001.
4. H. Kimura and S. Kobayashi. An Analysis of Actor/Critic Algorithm Using Eligibility Traces: reinforcement learning with imperfect value functions, *Proceedings of the ICML-98,* 1998.
5. V. R. Konda an d J. N. Tsitsiklis. Actor-Critic Algorithms. *SIAM Journal on Control and Optimization,* Vol. 42, No. 4:1143-1166, 2003.
6. R. Munos. Policy Gradient in Continuous Time. *Journal of Machine Learning Research 7,* pp. 771-791, 2006.
7. J. Peters, S. Vijayakumar, and S. Schaal. Reinforcement learning for humanoid robotics. *Humanoids2003, 3rd IEEE-RAS International Conference on Humanoid Robots.* Karlsruhe, Germany, Sept.29-30, 2003.
8. R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction,* MIT Press, Cambridge, MA, 1998.
9. C. Watkins and P. Dayan. Q-Learning. *Machine Learning,* vol. 8:279–292, 1992.
10. R. Williams. Simple Statistical Gradient Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning,* vol. 8:299–256, 1992.

Derivation of Equation 1

π has been defined as the normal distribution $N(\theta, \sigma_a^2)$. Therefore

$$\pi(a_t; s_t, \theta) = \frac{1}{\sqrt{2\pi}\sigma_a} \exp\left(-\frac{1}{2\sigma_a^2}(a_t - \theta)^2\right) \quad \text{and} \quad \frac{\partial \ln \pi(a_t; s_t, \theta)}{\partial \theta^T} = \frac{1}{\sigma_a^2}(a_t - \theta).$$

Consequently

$$\widehat{g} = \left(\sum_{t=0}^{T-1} \delta a_t + y - c\right)\left(\sum_{t=0}^{T-1} \frac{1}{\sigma_a^2}(a_t - \theta)\right).$$

Let us define

$$\xi = \sum_{t=0}^{T-1} \delta(a_t - \theta) = s_T - \theta.$$

Obviously

$$\widehat{g} = (\xi + y + (\theta - c))\left(\frac{1}{\delta\sigma_a^2}\xi\right).$$

ξ is a sum of independent random variables. Its distribution is easy to derive as $N(0, \delta\sigma_a^2)$. Note that ξ and s_T have the same variance. Since for each normal random variable X, the equality $\mathcal{E}(X - \mathcal{E}X)^4 = 3(\mathcal{V}X)^2$ holds, we obtain

$$\begin{aligned}
\mathcal{V}\widehat{g} &= \mathcal{E}\left(\big(\xi + y + (\theta - c)\big)\xi/\delta\sigma_a^2 - 1\right)^2 \\
&= \mathcal{E}\left(\frac{\xi^4 + 2\xi^3 y + 2\xi^3(\theta - c) + y^2\xi^2 + 2y(\theta - c)\xi^2 + (\theta - c)^2\xi^2}{\delta^2\sigma_a^4} - 2\frac{\xi^2 + y\xi + (\theta - c)\xi}{\delta\sigma_a^2} + 1\right) \\
&= \frac{3\delta^2\sigma_a^4 + \delta\sigma_a^2\sigma_y^2 + (\theta - c)^2\delta\sigma_a^2}{\delta^2\sigma_a^4} - 2\frac{\delta\sigma_a^2}{\delta\sigma_a^2} + 1 = 2 + \frac{1}{\delta\sigma_a^2}(\sigma_y^2 + (\delta - c)^2).
\end{aligned}$$

■