Recursive Adaptation of Stepsize Parameter for Unstable Environments

(Anonumous Author)

ABSTRACT

In this article, we propose a method to adapt stepsize parameters used in reinforcement learning for dynamic environments. In general reinforcement learning situations, a stepsize parameter is decreased to zero during learning, because the environment is generally supposed to be noisy but stable, such that the true expected rewards are fixed. On the other hand, we assume that in the real world, the true expected reward changes over time and hence, the learning agent must adapt the change through continuous learning. We derive the higher-order derivatives of exponential moving average (which is used to estimate the expected values of states or actions in major reinforcement learning) using stepsize parameters. We also illustrate a mechanism to calculate these derivatives in a recursive manner. Using the mechanism, we construct a precise and flexible adaptation method for the stepsize parameter in order to minimize square errors or maximize a certain criterion. The proposed method is validated both theoretically and experimentally.

Keywords

Reinfocement Learning, Adaptation of Learning Parameter, Dynamic Environment

1. INTRODUCTION

In most of the works on reinforcement learning that are used in agent learning, it is supposed that the environment for agents is stable during and after learning. In other words, while the environment may react to agents' action and provide rewards dynamically, the rules of the change and the mechanisms of rewarding are supposed to be stable forever. In such a case, it is reasonable that a stepsize parameter α is monotonically decreased to 0 through learning in the following temporal difference(TD) learning algorithm in order to estimate the expected values of the states or actions (Q-value) [6].

$$Q_{t+1}(state_t, act_t) = (1-\alpha)Q_t(state_t, act_t) + \alpha(r_t + \gamma \max_{act'} Q_t(state_{t+1}, act')) \quad (1)$$

By decreasing α sufficiently, we can reduce the noisy factors included in state transitions and rewarding errors. After the Q-values seem to be sufficiently near the true expected values, the agents generally stop learning and behave on the basis of the fixed Q-value. An important assumption here is that the true expected values are constant during and after learning [3].

On the other hand, in common real world problems, especially the problems on open and multiagent systems, the environment may change gradually or rapidly. For example, market systems such as the stock market and foreign exchange can be affected by both agents' behavior and various other fundamental conditions. Therefore, it is difficult to suppose that the true expected rewards of states or actions are stable. Instead, agents in such an environment should continue learning to adapt to changes in the environments. In this case, since we cannot decrease the stepsize parameter α monotonically, we control it such that it is capable of meeting the changes in the environment.

In order to adapt to such dynamic and unstable environments, [4] proposed a method, called optimal stepsize algorithm (OSA), to control stepsize parameters in order to minimize noise factors on the basis of the relationships among the stepsize parameter, noise variance, and changes in learning values. [5] also proposed a framework to accumulate error variance to find out the suitable learning parameters. In both works, the focus is only on minimizing estimation errors, which the effect of the changes in the stepsize parameter on the learning processes is ignored.

For this issue, we focus on the effects of the changes in the stepsize parameter on the learning process, and extend the learning process to estimate the effects. On the basis of the estimation, we can construct a method to adjust the stepsize parameter in order to optimize a certain criteria, for example, minimizing an error.

2. EXPONENTIAL MOVING AVERAGE AND STEPSIZE PARAMETER

2.1 Exponential Moving Average

In reinforcement learning, for example, TD learning, an agent learns to estimate the expected value of each state or action that is used in decision making according to the rewards that the agent receives as results of his/her action in the unknown environment. Generally, the estimation is done by the following *exponential moving average* (EMA) equation.

$$\tilde{x}_{t+1} = (1-\alpha)\tilde{x}_t + \alpha x_t, \quad (2)$$

Cite as: Recursive Adaptation of Stepsize Parameter for Unstable Environments, Anonymous Author, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. XXX-XXX.

where x_t and $\tilde{x}_t$ are the actual observed value (for example, received reward r_t) and the corresponding expected value, respectively, that are updated through discrete time line t. α is a *stepsize parameter*, which indicates whether the agent regards recent observed values x_t as important, or the agent should take a long-term average so as to calculate the true expected value ($\tilde{x}_t$). It is known that $\tilde{x}_t$ can be interpreted to be an approximation of a moving average of x_t in the following time-window:

$$T \quad = \quad \frac{2}{\alpha} - 1. \tag{3}$$

2.2 Best Follow-up to Random Walk

Suppose that an observation sequence $\{x_t\}$ consists of a true value sequence $\{s_t\}$ and noise sequence $\{\epsilon_t\}$ as described in the following equation.

$$x_t \quad = \quad s_t + \epsilon_t, \tag{4}$$

where ϵ_t is a random noise with average 0 and standard deviation σ_ϵ, and is independent from s_t. Furthermore, suppose that the true-value sequence $\{s_t\}$ is a random walk sequence as defined by the equation

$$s_{t+1} \quad = \quad s_t + v_t,$$

where v_t is a random value with average 0 and standard deviation σ_v.

In this case, we can derive the following lemma and theorem.

Lemma 1.
The mean square error $\mathbf{E}(\delta_t^2) = \mathbf{E}((\tilde{x}_t - x_t)^2)$ of expected value $\tilde{x}_t$ acquired by eq. (2) using observation x_t that follows eq. (4) is given by the following equation.

$$\mathbf{E}(\delta_t^2) \quad = \quad \frac{1}{2-\alpha}(2\sigma_\epsilon^2 + \frac{1}{\alpha}\sigma_v^2). \tag{5}$$

(See section A for the proof.) □

Theorem 1.
The stepsize parameter α that minimizes the mean square error $\mathbf{E}(\delta_t^2)$ is given by the following equation.

$$\alpha \quad = \quad \frac{-\gamma^2 + \sqrt{\gamma^4 + 4\gamma^2}}{2}, \tag{6}$$

where $\gamma = \frac{\sigma_v}{\sigma_\epsilon}$.
(See section B for the proof.) □

The theorem says that, if observed values consist of random walk values and independent random noise, the best stepsize parameter to balance the follow-up to the random walk and smoothening so as to remove the noise factor can be determined by eq. (6).

2.3 Recursive Exponential Moving Average and Higher-Order Partial Derivatives

In order to determine the stepsize parameter using eq. (6), the agent needs to know the standard deviations of the random walk and noise factor. In general, however, in real learning applications, these values are not known or change over time. Therefore, we try to extract the derivatives of the expected value $\tilde{x}_t$ using the stepsize parameter α, and construct a method to adapt α according to a given sequence of observation $\{x_t\}$.

First, we introduce the following *recursive exponential moving average* (REMA) $\xi_t^{\langle k\rangle}$ by applying eq. (2) recursively:

$$\begin{aligned}
\xi_t^{\langle 0\rangle} &= x_t \\
\xi_{t+1}^{\langle 1\rangle} &= \tilde{x}_{t+1} = (1-\alpha)\tilde{x}_t + \alpha x_t \\
\xi_{t+1}^{\langle k\rangle} &= \xi_t^{\langle k\rangle} + \alpha(\xi_t^{\langle k-1\rangle} - \xi_t^{\langle k\rangle}) \\
&= (1-\alpha)\xi_t^{\langle k\rangle} + \alpha\xi_t^{\langle k-1\rangle} \\
&= \alpha\sum_{\tau=0}^{\infty}(1-\alpha)^\tau \xi_{t-\tau}^{\langle k-1\rangle}.
\end{aligned} \tag{7}$$

With regard to REMA, we can state the following lemma and theorem.

Lemma 2.
The first partial derivative of REMA $\xi_t^{\langle k\rangle}$ by α is given by the following equation:

$$\frac{\partial \xi_t^{\langle k\rangle}}{\partial\alpha} \quad = \quad \frac{k}{\alpha}(\xi_t^{\langle k\rangle} - \xi_t^{\langle k+1\rangle}). \tag{8}$$

(See section C for the proof.) □

Theorem 2.
The k-th partial derivative of EMA $\tilde{x}_t$ ($= \xi_t^{\langle 1\rangle}$) is given by the following equation:

$$\frac{\partial^k \tilde{x}_t}{\partial\alpha^k} \quad = \quad (-\alpha)^{-k}k!(\xi_t^{\langle k+1\rangle} - \xi_t^{\langle k\rangle}). \tag{9}$$

(See section D for the proof.) □

2.4 Gradient Descent Adaptation of Stepsize Parameter Using Higher-order Derivatives and REMA

Because theorem 2 provides the derivatives of $\tilde{x}_t$ by α, we can construct algorithms to optimize a certain criterion, for example, mean square errors, by gradient descent/ascent methods. An important aspect of theorem 2 is that it can provide derivatives of any order. Therefore, we can form more precise gradient descent/ascent methods. We refer to such methods that use higher-order derivatives given by REMA as *recursive adaptation of stepsize parameters* (RASP).

Suppose that $\Delta\tilde{x}_t$ is the change in $\tilde{x}_t$ when α changes by $\Delta\alpha$. In this case, $\Delta\tilde{x}_t$ can be represented by Taylor expansion and theorem 2 as follows:

$$\begin{aligned}
\Delta\tilde{x}_t &= \sum_{k=1}^{\infty}\frac{1}{k!}\frac{\partial^k\tilde{x}_t}{\partial\alpha^k}\Delta\alpha^k \\
&= \sum_{k=1}^{\infty}(-1)^k\left(\frac{\Delta\alpha}{\alpha}\right)^k(\xi_t^{\langle k+1\rangle} - \xi_t^{\langle k\rangle}).
\end{aligned} \tag{10}$$

Further, generally, $\Delta\xi_t^{\langle k\rangle}$ for any k can be estimated by the first Tailor expansion and lemma 2 as follows: [1]

$$\Delta\xi_t^{\langle k\rangle} \quad = \quad \Delta\alpha\frac{\partial\xi_t^{\langle k\rangle}}{\partial\alpha} \tag{11}$$

$$\simeq \quad k\left(\frac{\Delta\alpha}{\alpha}\right)(\xi_t^{\langle k\rangle} - \xi_t^{\langle k+1\rangle}). \tag{12}$$

These expansions indicate that RASP exhibits the following features.

[1]We can also use a higher-order Tailor expansion to utilize higher-order derivatives as shown in the appendix.

1. We can approximate the precise changes in the estimation value $\tilde{x}_t$ even for a large $\Delta\alpha$, using higher-order derivatives calculated by REMA. Therefore, we can change α rapidly.

2. We can also calculate $\Delta\xi_t^{\langle k\rangle}$ by a modification of α, using the derivatives of $\xi_t^{\langle k\rangle}$. Therefore, the values of the variables that are affected by the changes in α are kept precise.

Of course, it is impossible to calculate infinite higher-order derivatives. Instead, we can set upper limit of k large enough to achieve the required precision. Because the calculation of REMA itself is very simple, the cost to calculate higher-order derivatives is small.

The following procedure details the use of RASP to minimize the square error between the expected value $\tilde{x}_t$ and the actual observation x_t. (We call this procedure RASP-MSE.)

> Initialize: $\forall k \in \{0 \ldots k_{\max}-1\} : \xi^{\langle k\rangle} \leftarrow x_0$
> **while** forever **do**
>   Let x be an observation.
>   **for** $k = k_{\max}-1$ to 1 **do**
> $\xi^{\langle k\rangle} \leftarrow (1-\alpha)\xi^{\langle k\rangle} + \alpha\xi^{\langle k-1\rangle}$
>   **end for**
> $\xi^{\langle 0\rangle} \leftarrow x$
> $\delta \leftarrow \xi^{\langle 1\rangle} - x$
>   Calculate $\frac{\partial\xi^{\langle 1\rangle}}{\partial\alpha}$ by eq. (9).
>   **for** $k = 1$ to $k_{\max}-1$ **do**
>   Calculate $\Delta\xi^{\langle k\rangle}$ by eq. (10) and eq. (12).
> $\xi^{\langle k\rangle} \leftarrow \xi^{\langle k\rangle} + \Delta\xi^{\langle k\rangle}$
>   **end for**
>   calculate a new α according to δ and $\frac{\partial\xi^{\langle 1\rangle}}{\partial\alpha}$.
> **end while**

In this procedure, there are several possible ways to decide the value of $\Delta\alpha$. As in a general gradient descent method, in this case, the only restriction is that $\Delta\alpha < 0$ when $\delta\frac{\partial\xi^{\langle 1\rangle}}{\partial\alpha} > 0$, and $\Delta\alpha > 0$ otherwise. In addition, because of the nature of EMA, we should keep the following points in mind.

- α should be a real number in $[0, 1]$.
- α should not get too close to 0 because eq. (9) has a singular point at $\alpha = 0$.

Therefore, in the experiments described below, we use the following procedure to decide $\Delta\alpha$.

$$
\begin{aligned}
\gamma'_{\text{old}} &\leftarrow \sqrt{\frac{\alpha^2}{1-\alpha}} \\
\lambda &\leftarrow -\bar{\lambda}\cdot\text{sign}(\delta\frac{\partial\xi^{\langle 1\rangle}}{\partial\alpha}) \\
\gamma'_{\text{new}} &\leftarrow \exp(\log(\gamma'_{\text{old}}) + \lambda) \\
\alpha_{\text{new}} &\leftarrow \frac{-\gamma'^2_{\text{new}} + \sqrt{\gamma'^4_{\text{new}} + 4\gamma'^2_{\text{new}}}}{2} \\
\Delta\alpha &\leftarrow \alpha_{\text{new}} - \alpha
\end{aligned}
$$

In this procedure, α is modified according to the uniformed-step changes in the logarithmic value of γ in eq. (6). Therefore, the changes in α are large when α is around 0.5, and small when α is close to 0 or 1.

3. EXPERIMENTS

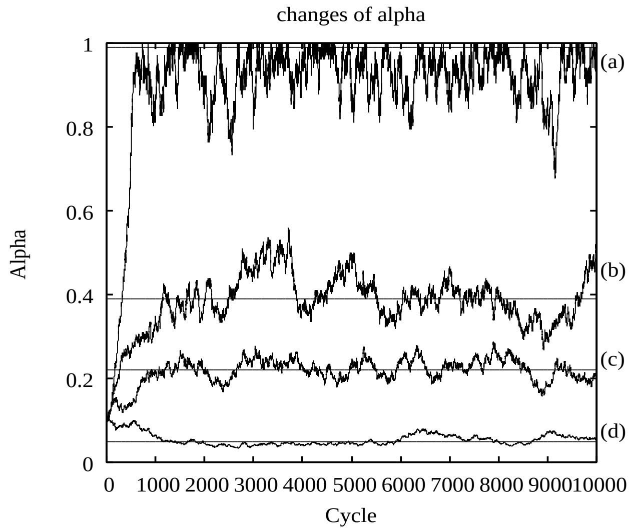


Figure 1: Exp.1: Changes in α through the Learning of Observed Value Using the Various Ratios of Standard Deviations of Random Walk and Noise (γ).

3.1 Exp.1: Learning Best α for Noise Reduction

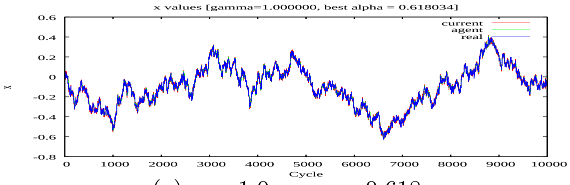
In the first experiment, we show that the above procedure to adapt α yields the best stepsize parameter value for noise reduction that is determined by eq. (6).

Figure 1 shows the results of the adaptation of α through the learning of observation sequences $\{x_t\}$ with various γ (the ratio of standard deviations of random walk and noise). In each case of this experiment, we use the following standard deviations of random walk and noise.

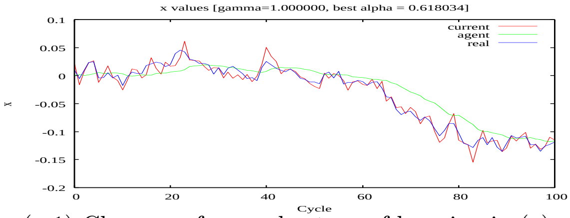
	σ_s	σ_ϵ	(γ)	(α_{best})
(a)	0.01	0.001	10.00	0.990
(b)	0.01	0.020	0.50	0.390
(c)	0.01	0.040	0.25	0.221
(d)	0.01	0.200	0.05	0.048

Each curve in the graph of Figure 1 shows the changes in α through the learning of expected value $\tilde{x}_t$ by eq. (2) and adaptation of α by RASP-MSE. The horizontal axis in the graphs indicates the learning (and adaptation) cycle, while the vertical axis represents the value of α. Further, the horizontal line in each graph indicates the best stepsize parameter (α_{best}) as calculated by eq. (6). As shown in these graphs, α approaches the best value and is then consistent through learning. Note that α does not converge to the best value because of the noise factors added in the observed value. Fortunately, the perturbation is large only when α is relatively large; in this case, the effect of α changes slowly, so that the behavior of the learning does not change drastically even α changes with a large step.

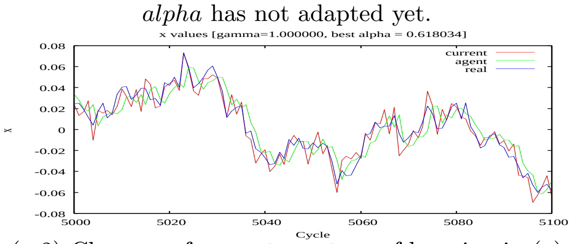
Figure 2 shows the changes in the expected value $\tilde{x}_t$ as calculated by EMA. In the figure, (a) and (b) are the cases where the parameter γ is almost equal to 1.0 and 0.1, respectively. In other words, (a) is the case where the standard deviation of the true random walk value s_t exceeds the standard deviation of noise ϵ_t sufficiently, and (b) is the case where the noise factor is larger than the random walk. Graphs (a-1) and (a-2) show detailed close-ups of the



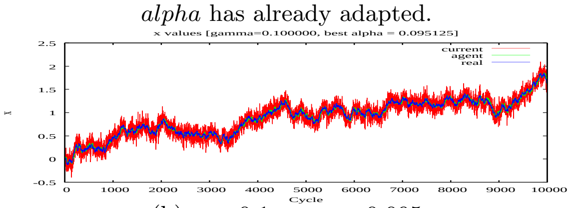
(a) $\gamma = 1.0, \alpha_{\text{best}} = 0.618$



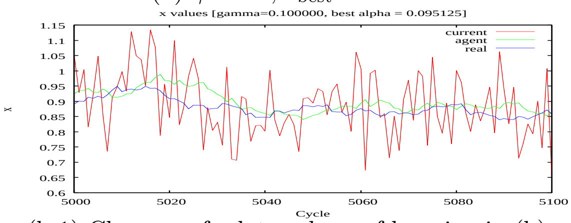
(a-1) Close-up of an early stage of learning in (a) *alpha* has not adapted yet.



(a-2) Close-up of an mature stage of learning in (a) *alpha* has already adapted.



(b) $\gamma = 0.1, \alpha_{\text{best}} = 0.095$



(b-1) Close-up of a later phase of learning in (b)

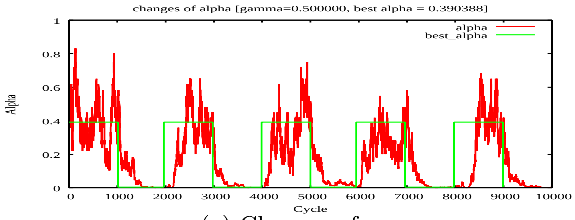
Figure 2: Exp.1: Changes in the Expected Value $\tilde{x}_t$.

changes in (a) at the early and mature stages [2] of learning, respectively. Similarly, (b-1) shows the detailed changes taking place at the mature stage of learning in (b) in detail. In these graphs, (a-1) shows that the expected value $\tilde{x}_t$ can not follow the quick changes in the true value s_t but does smoothen the changes. This is so as α is still too small at the early stage of learning. On the other hand, in (a-2), α is adapted to be suitable, and consequently, $\tilde{x}_t$ can follow s_t with minimum delay. In (b) and (b-1), α is kept small enough to reduce the large noise factor and allow $\tilde{x}_t$ to yield the best estimate of s_t.

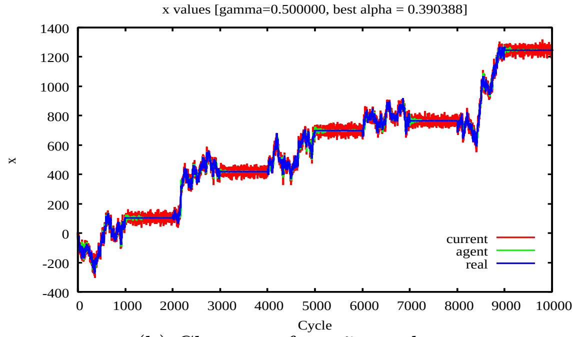
As shown in these results, RASP-MSE can acquire the suitable stepsize parameter α for a given sequence.

3.2 Exp.2: The case of Square-Waved γ

[2]Here, "mature" stage implies a phase when the learning is almost complete and α is close enough to the optimal value.



(a) Changes of α



(b) Changes of x_t, $\tilde{x}_t$, and s_t

Figure 3: Exp.2: Square-waved γ.

In order to show how RASP-MSE can follow the changes in the environment, we conducted an experiment in which γ changes along a square wave over time.

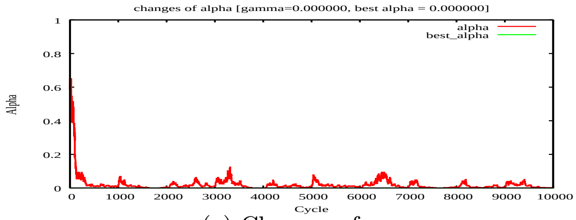
Figure 3 shows the result of an experiment to adapt α by RASP-MSE in the EMA learning of $\tilde{x}_t$ when γ alternates between 0.5 and 0.0005 every 1000 steps. The graph in (a) shows the changes in α through learning (a curve) along with the ideal changes expected according to γ (a square wave). The top and the bottom of the square wave are 0.39 and 0.0005, respectively. As shown in this graph, α tries to follow the changes in γ. Graph (b) shows the changes in the expected value $\tilde{x}_t$, observed value x_t, and true value s_t. During the period when γ is small (0.0005, where α's ideal value is 0.0005), $\tilde{x}_t$ becomes a type of long-term moving average so as to reduce the large noise factor: On the other hand, $\tilde{x}_t$ follows x_t tightly during the period when γ is large (0.5, where α's ideal value is 0.39). This result shows that RASP-MSE can follow the changes in the environment and determine a suitable stepsize parameters.

3.3 Exp.3: Square-waved True Value

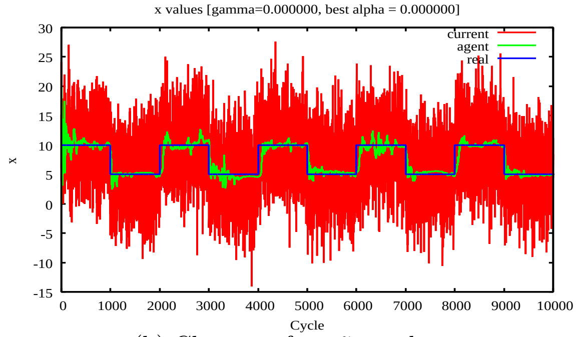
EMA is used in general reinforcement learning, for example, eq. (1), because it can reduce noise and yield a value that approaches the stable true value. In the second experiment, we suppose that the true value is almost stable but does change occationally. In such a case, the learning mechanism needs to detect the changes in the true value. In the actual experiment, we use a sequence of true values $\{s_t\}$ that follows a square wave over time.

Figure 4 shows the result of an experiment to adapt α by RASP-MSE in the EMA learning of $\tilde{x}_t$ when the true value s_t alternates between 0.0 and 0.5 every 1000 steps. In this experiment the standard deviation of noise ϵ_t is 5.0. (a) shows the changes in α, and (b), in x_t, $\tilde{x}_t$, and s_t through learning. (c) shows a result of the case that we apply OSA [4] to the same problem for the comparison.

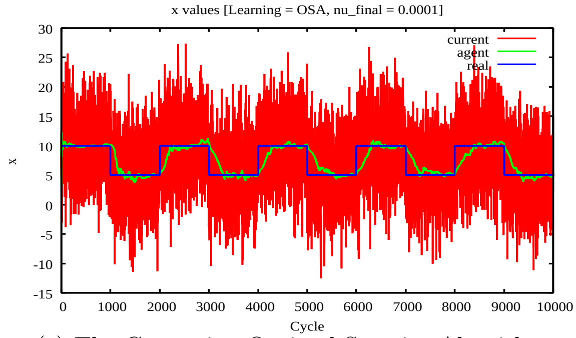
(b) indicates that RASP-EMA reduces the large noise factor and at the same time can follow the changes in the true



(a) Changes of α



(b) Changes of x_t, $\tilde{x}_t$, and s_t.



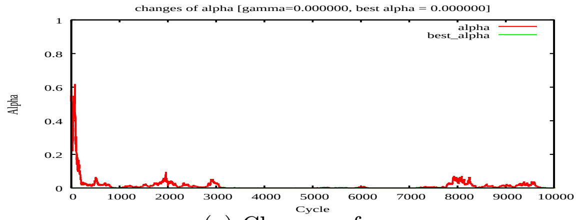
(c) The Case using Optimal Stepsize Algorithm

Figure 4: Exp.3: Learning Square-waved True Value s_t

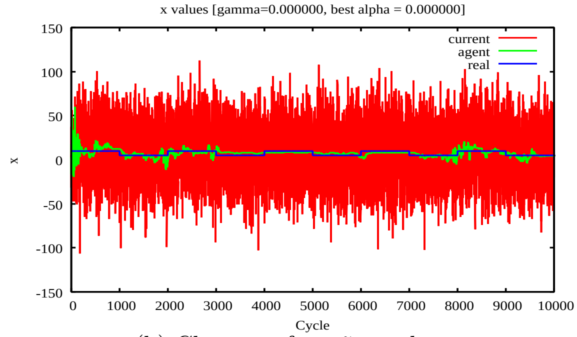
value. Compared with (c), we found that following the true value is more precise by RASP-EMA than by OSA. Actually, the average square error of $\tilde{x}_t$ from x_t in (b) is 1.192, while the error in (c) is 2.496. Corresponding changes in α in (a) shows that α approaches zero almost all the times but is relatively large at the time when the true value s_t changes ($t = 1000, 2000, \ldots$). From the meaning of α in EMA ($\tilde{x}_t$ follows the previous observed value x_t when α is large, and $\tilde{x}_t$ becomes a long-term moving average of x_t when α is small), the change in α shown in (a) indicates that RASP-MSE detects the timing of changes in s_t and lets an agent regard the recent observation as plausible: On the other hand, RASP-MSE lets the agent use the long-term smoothed value when the environment is stable. In other words, RASP-MSE can control the features of learning by EMA in accordance with the changes in the environment.

3.3.1 Limitations of RASP-MSE with Regard to Square-Waved True Value Sequences

Exp.3, described in section 3.3, shows the ability of RASP-MSE to follow a square-waved true value sequence. However, the proposed procedure is not able to follow any square waves. For example, if the observed value x_t includes noise with standard deviation 30.0 instead of 5.0, the RASP-MSE



(a) Changes of α



(b) Changes of x_t, $\tilde{x}_t$, and s_t.

Figure 5: Exp.3-2: Learning in the Case of Small Square-waved True Value s_t

does not suitably follow the change in the true value s_t but regards the change as noise (figure 5). In the graph, $\alpha \cong 0$ during steps 3000–7000 steps. This implies that $\tilde{x}_t$ becomes a long-term moving average of the observation, where the term of the average is longer than the cycles of changes in the true value.

We can derive the theoretical upper-limit of the adaptation to the changes in the small square-waved true value as follows:

Suppose that the true value s_t changes according to the following formula:

$$s_t = \begin{cases} -\delta & : \quad (2n-1)T \leq t < 2nT \\ \delta & : \quad 2nT \leq t < (2n+1)T \end{cases},$$

where $2T$ is a cycle of square-waved changes in the true value.

If α is almost zero such that $\tilde{x}_t$ represents a long-term moving average of x_t, the mean square error E_0 of the expectation is as follows:

$$\begin{aligned} E_0 &= \mathbf{E}((x_t - \mathbf{E}(x_t))^2) \\ &= \delta^2 + \sigma_\epsilon^2. \end{aligned}$$

On the other hand, suppose that we can control α optimally, that is, α is raised to 1 at $t = nT$, and is decayed to be the value $1/(1 + t - nT)$ otherwise. In this case, $\tilde{x}_t$ becomes an average of x_t during each half cycle. Therefore, the mean square error E_{opt} is as follows:

$$\begin{aligned} E_{opt} &= \mathbf{E}((x_t - \tilde{x}_t)^2) \\ &= \frac{1}{T}\left[4\delta^2 + \sigma_\epsilon^2 + \sum_{\tau=1}^{T}(\sigma_\epsilon^2 + \frac{\sigma_\epsilon^2}{\tau})\right] \\ &= \frac{1}{T}\left[4\delta^2 + T\sigma_\epsilon^2 + \sigma_\epsilon^2\mathcal{H}_T,\right], \end{aligned}$$

where $\mathcal{H}_T = \sum_{\tau=1}^{T}\frac{1}{\tau}$ is a harmonic series.

If $E_0 < E_{opt}$, we obtain the following inequality:

$$(T-4)\delta^2 \quad < \quad \mathcal{H}_T \sigma_\epsilon^2. \tag{13}$$

This is satisfied when $T \leq 4$. This implies that it is impossible to follow this quick changes ($T \leq 4$) by the proposed procedure, because the long-term average (the case of $\alpha \sim 0$) provides better estimation than the expectation by EMA with adaptive α.

In the case of $T > 4$, eq. (13) can be written as follows:

$$\frac{\delta^2}{\sigma_\epsilon^2} \quad < \quad \frac{\mathcal{H}_T}{T-4}. \tag{14}$$

This inequality shows the limitation of EMA with adaptive stepsize parameters: When the changes in the true value (δ) is small, the noise (σ_ϵ) is large, and/or the interval time (T) is short as shown in eq. (14), then it is impossible to follow the true value by EMA.

Consider the case of the experiment shown in figure 5, where $\delta = 5/2$, $\sigma_\epsilon = 30.0$, and $T = 1000$. Therefore, the left and right hand sides of eq. (14) are 0.0833 and 0.0867, respectively. This means that the condition of this experiment is beyond the scope of the EMA shown by eq. (14). This is the reason why RASP-MSE failed to adapt α in this experiment. As shown by the actual values on both sides of the inequality, however, the condition is very close to the boundary. Therefore, RASP-MSE sometimes detects the changes in the true value as shown in graph (a) of figure 5.

4. DISCUSSION AND SUMMARY

In this article, we derive the relations between stepsize parameter α and expected value $\tilde{x}_t$ acquired by EMA, and provide a method called RASP that calculates the higher-order derivatives of $\tilde{x}_t$ by α. We also propose a procedure called RASP-MSE that adjusts α suitably for given observed data both to reduce noise factors in the observation and to follow the changes in the environment. Experiments illustrate the functionality and performance of RASP-MSE for adjusting the stepsize parameters as shown in theorems and lemmas.

The main feature of RASP is that we can obtain derivatives $\partial\tilde{x}_t/\partial\alpha$. Therefore, we can apply it to various optimization applications that require EMA. For example, it can not only be applied to situations where the minimization of estimation error is desired, but also to the learning of decision making directly, for example, back-propagations in neural networks. Thus, it can be said that RASP has more potential than the other adaptation mechanisms of stepsize parameters such as OSA [4].

The stochastic gradient adaptive (SGA) stepsize method [1, 2] is identical to RASP-MSE if we use only the first-order derivative. As we can calculate higher-order derivatives, adaptation can be more quick and precise.

There still several open issues that include:

- To apply RASP-MSE to TD learning and multiagent learning, which may not follow the assumption of random walk.
- To utilize higher-order derivatives to calculate the best stepsize instead to change it gradually.

5. REFERENCES

[1] A. Benveniste, M. Metivier, and P. Priouret. *Adaptive Algorithms and Stockastic Approximations*. Springer, Dec. 1990.

[2] S. C. Douglas and V. J. Mathews. Stochastic gradient adaptive step size algorithms for adaptive filtering. In *Proc. International Conference on Digital Signal Processing*, pages 142–147, 1995.

[3] E. Even-dar and Y. Mansour. Learning rates for q-learning. *Journal of Machine Learning Research*, 5:2003, Dec. 2003.

[4] A. P. Gorge and W. B. Powell. Adaptive stepsizes for recursive estimation with applications in approximate dynamic programming. *Machine learning*, 65(1):167–198, 2006.

[5] M. Sato, H. Kimura, and S. Kobayashi. TD algorithm for the variance of return and mean-variance reinforcement learning (in japanese). *Transactions of the Japanese Society for Artificial Intelligence*, Vol. 16(No. 3F):353–362, 2001.

[6] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

APPENDIX

A. PROOF OF LEMMA 1

Suppose that the observed value x_t follows eq. (4). In this case, the expected value $\tilde{x}_t$ calculated by eq. (2) can be written as follows:

$$\begin{aligned}
\tilde{x}_{t+1} &= \alpha\sum_{\tau=0}^{\infty}(1-\alpha)^\tau x_{t-\tau} \\
&= \alpha\sum_{\tau=0}^{\infty}(1-\alpha)^\tau s_t - \alpha\sum_{\tau=0}^{\infty}(1-\alpha)^\tau\sum_{\tau'=1}^{\tau} v_{t-\tau'} \\
&\quad +\alpha\sum_{\tau=0}^{\infty}(1-\alpha)^\tau \epsilon_{t-\tau}.
\end{aligned}$$

Here, we can rearrange the second term according to τ', and obtain the following equation:

$$\tilde{x}_{t+1} \quad = \quad s_t - \sum_{\tau'=1}^{\infty}(1-\alpha)^{\tau'} v_{t-\tau'} + \alpha\sum_{\tau=0}^{\infty}(1-\alpha)^\tau \epsilon_{t-\tau}.$$

Therefore, the estimation error δ_t (difference between the expected and observed values) becomes

$$\begin{aligned}
\delta_t &= -\sum_{\tau'=1}^{\infty}(1-\alpha)^{\tau'} v_{t-1-\tau'} \\
&\quad +\alpha\sum_{\tau=0}^{\infty}(1-\alpha)^\tau \epsilon_{t-1-\tau} - v_{t-2}.
\end{aligned}$$

Because ϵ_t and v_t are independent random numbers with means 0 and standard deviations σ_ϵ^2 and σ_v^2, respectively, the mean square of the above error $\mathbf{E}(\delta_t^2)$ can be calculated as follows:

$$\mathbf{E}(\delta_t^2) \quad = \quad \frac{1}{2-\alpha}(2\sigma_\epsilon^2 + \frac{1}{\alpha}\sigma_v^2).$$

∎

B. PROOF OF THEOREM 1

The derivative of mean square error $\mathbf{E}(\delta_t^2)$ by α is as follows:

$$\begin{aligned}\frac{\partial \mathbf{E}(\delta_t^2)}{\partial \alpha} &= \frac{1}{(2-\alpha)^2}(2\sigma_\epsilon^2 + \frac{1}{\alpha}\sigma_v^2) + \frac{1}{2-\alpha}(-\frac{1}{\alpha^2}\sigma_v^2) \\ &= \frac{2(\alpha^2\sigma_\epsilon^2 + (\alpha-1)\sigma_v^2)}{\alpha^2(2-\alpha)^2}.\end{aligned}$$

Suppose that the above derivative is equal to 0. Then, we can obtain a solution of α in the range $(0, 1)$ as follows:

$$\alpha = \frac{-\sigma_v^2 + \sqrt{\sigma_v^4 + 4\sigma_\epsilon^2\sigma_v^2}}{2\sigma_\epsilon^2} = \frac{-\gamma^2 + \sqrt{\gamma^4 + 4\gamma^2}}{2}.$$

■

C. PROOF OF LEMMA 2

First, we show the following lemma.

Lemma 3.

$$\xi_{t+1}^{\langle k\rangle} = \alpha^2 \sum_{\tau=0}^{\infty} \tau(1-\alpha)^{\tau-1}\xi_{t-\tau}^{\langle k-2\rangle}. \qquad (15)$$

□

Proof
Suppose that

$$\begin{aligned}\eta_{t+1} &= \alpha^2 \sum_{\tau=0}^{\infty} \tau(1-\alpha)^{\tau-1}\xi_{t-\tau}^{\langle k-2\rangle} \\ &= \alpha^2 \Big[1(1-\alpha)^0\xi_{t-1}^{\langle k-2\rangle} + 2(1-\alpha)^1\xi_{t-2}^{\langle k-2\rangle} \\ &\qquad + 3(1-\alpha)^2\xi_{t-3}^{\langle k-2\rangle} + \ldots\Big].\end{aligned}$$

Then, we can obtain the following equation:

$$\begin{aligned}(1-\alpha)\eta_t &= \alpha^2 \Big[1(1-\alpha)^1\xi_{t-2}^{\langle k-2\rangle} + 2(1-\alpha)^2\xi_{t-3}^{\langle k-2\rangle} \\ &\qquad + 3(1-\alpha)^3\xi_{t-4}^{\langle k-2\rangle} + \ldots\Big].\end{aligned}$$

This can be rewritten as follows:

$$\begin{aligned}\eta_{t+1} - (1-\alpha)\eta_t &= \alpha^2 \Big[(1-\alpha)^0\xi_{t-1}^{\langle k-2\rangle} + (1-\alpha)^1\xi_{t-2}^{\langle k-2\rangle} \\ &\qquad + (1-\alpha)^2\xi_{t-3}^{\langle k-2\rangle} + \ldots\Big] \\ &= \alpha^2 \sum_{\tau=0}^{\infty}(1-\alpha)^\tau \xi_{t-1-\tau}^{\langle k-2\rangle} \\ &= \alpha\xi_t^{\langle k-1\rangle}.\end{aligned}$$

Finally, we can obtain the recurrence formula:

$$\eta_{t+1} = (1-\alpha)\eta_t + \alpha\xi_t^{\langle k-1\rangle}.$$

This formula is the same as the one for $\xi_t^{\langle k\rangle}$. Therefore, if $\eta_0 = \xi_0^{\langle k\rangle}$, η_t is identical to $\xi_t^{\langle k\rangle}$ for all t. Therefore, we can obtain eq. (15). ■

Using this lemma, we can prove Lemma 2 as follows:

In the case of $k = 1$, we can obtain the following equation:

$$\begin{aligned}\frac{\partial \xi_t^{\langle 1\rangle}}{\partial \alpha} &= \frac{\partial \tilde{x}_t}{\partial \alpha} \\ &= \frac{\partial}{\partial \alpha}\left[\alpha\sum_{\tau=0}^{\infty}(1-\alpha)^\tau x_{t-\tau-1}\right] \\ &= \sum_{\tau=0}^{\infty}(1-\alpha)^\tau x_{t-\tau-1} \\ &\quad + \alpha\sum_{\tau=0}^{\infty}(-1)\tau(1-\alpha)^{\tau-1}x_{t-\tau-1} \\ &= \frac{1}{\alpha}(\xi_t^{\langle 1\rangle} - \xi_t^{\langle 2\rangle}).\end{aligned}$$

Therefore, eq. (8) is satisfied when $k = 1$.

Suppose that eq. (8) is satisfied for any $k < k'$. Then, we can calculate the k'-th derivative as follows:

$$\begin{aligned}\frac{\partial \xi_t^{\langle k'\rangle}}{\partial \alpha} &= \frac{\partial}{\partial \alpha}\left[\alpha\sum_{\tau=0}^{\infty}(1-\alpha)^\tau \xi_{t-\tau-1}^{\langle k'-1\rangle}\right] \\ &= \frac{1}{\alpha}\xi_t^{\langle k'\rangle} - \frac{1}{\alpha}\xi_t^{\langle k'+1\rangle} \\ &\quad + \alpha\sum_{\tau=0}^{\infty}(1-\alpha)^\tau \frac{k'-1}{\alpha}(\xi_{t-\tau-1}^{\langle k'-1\rangle} - \xi_{t-\tau-1}^{\langle k'\rangle}) \\ &= \frac{k'}{\alpha}(\xi_t^{\langle k'\rangle} - \xi_t^{\langle k'+1\rangle}).\end{aligned}$$

As a result, eq. (8) holds for any $k > 0$. ■

D. PROOF OF THEOREM 2

In the case of $k = 1$, we can obtain the following equation:

$$\frac{\partial \tilde{x}_t}{\partial \alpha} = \frac{\partial}{\partial \alpha}\xi_t^{\langle 1\rangle} = (-\alpha)^{-1}(\xi_t^{\langle 1\rangle} - \xi_t^{\langle 2\rangle}).$$

Therefore, eq. (9) is satisfied when $k = 1$.

Suppose that eq. (9) is satisfied for any $k < k'$. Then, we can calculate the k'-th derivative as follows:

$$\begin{aligned}\frac{\partial^k \tilde{x}_t}{\partial \alpha^k} &= -(k-1)(-1)^{-(k-1)}\alpha^{-k}(k-1)!(\xi_t^{\langle k\rangle} - \xi_t^{\langle k-1\rangle}) \\ &\quad + (-1)^{-(k-1)}\alpha^{-(k-1)}\left[\frac{\partial}{\partial \alpha}\xi_t^{\langle k\rangle} - \frac{\partial}{\partial \alpha}\xi_t^{\langle k-1\rangle}\right]\end{aligned}$$

The first and second terms inside the brackets in the right hand side of this equation are $\frac{k}{\alpha}(\xi_t^{\langle k\rangle} - \xi k + 1_t)$ and $\frac{k-1}{\alpha}(\xi_t^{\langle k-1\rangle} - \xi k_t)$, respectively. Therefore,

$$\begin{aligned}\frac{\partial^k \tilde{x}_t}{\partial \alpha^k} &= (-1)^{-(k-1)}\alpha^{-k}(k-1)!\left[-k\xi_t^{\langle k+1\rangle} + k\xi_t^{\langle k\rangle}\right] \\ &= (-\alpha)^{-k}k!(\xi_t^{\langle k+1\rangle} - \xi_t^{\langle k\rangle}).\end{aligned}$$

As a result, eq. (9) holds for any $k > 0$. ■

Furthermore, the m-th derivatives of general REMA $\xi_t^{\langle k\rangle}$ by α can be shown using the same inductive method:

$$\frac{\partial^m \xi_t^{\langle k\rangle}}{\partial \alpha^m} = \frac{k}{\alpha^m}\sum_{i=0}^{m}(-1)^i \frac{m!}{i!(m-l)!}\frac{(k+i-1)!}{(k+i-m)!}\xi_t^{\langle k+i\rangle}.$$