



PERGAMON

Meta-learning in Reinforcement Learning

Nicolas Schweighofer^{a,*}, Kenji Doya^{a,b,1}

^aCREST, Japan Science and Technology Corporation, ATR, Human Information Science Laboratories, 2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan

^bATR Human Information Science Laboratories, 2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan

Received 6 September 2002; accepted 10 October 2002

Abstract

Meta-parameters in reinforcement learning should be tuned to the environmental dynamics and the animal performance. Here, we propose a biologically plausible meta-reinforcement learning algorithm for tuning these meta-parameters in a dynamic, adaptive manner. We tested our algorithm in both a simulation of a Markov decision task and in a non-linear control task. Our results show that the algorithm robustly finds appropriate meta-parameter values, and controls the meta-parameter time course, in both static and dynamic environments. We suggest that the phasic and tonic components of dopamine neuron firing can encode the signal required for meta-learning of reinforcement learning. © 2002 Elsevier Science Ltd. All rights reserved.

Keywords: Reinforcement learning; Dopamine; Dynamic environment; Meta-learning; Meta-parameters; Neuromodulation; TD error

1. Introduction

Reinforcement learning is a particularly attractive model of animal learning, as dopamine neurons have the formal characteristics of the teaching signal known as the temporal difference (TD) error (Schultz, 1998). However, crucial to successful reinforcement learning is the careful setting of several meta-parameters. We earlier proposed that these meta-parameters are set and adjusted by neuromodulator neurons (Doya, 2002). We introduce here a simple, yet robust, algorithm that not only finds appropriate meta-parameters, but also controls the time course of these meta-parameters in a dynamic, adaptive manner.

1.1. Reinforcement learning

The main issue in the theory of reinforcement learning (RL) is to maximize the long-term cumulative reward. Thus, central to reinforcement learning is the estimation of the value function

$$V(x(t)) = E \left[\sum_{k=0}^{\infty} \gamma^k r(t+k+1) \right],$$

where $r(t), r(t+1), r(t+2), \dots$ are the rewards acquired by following a certain action policy $x \rightarrow a$ starting from $x(t)$, and γ is a discount factor such that $0 < \gamma < 1$. The value function for the states before and after the transition should satisfy the consistency equation

$$V(x(t-1)) = E[r(t) + \gamma V(x(t))].$$

Therefore, any deviation from the consistency equation, expressed as

$$\delta(t) = r(t) + \gamma V(x(t)) - V(x(t-1)),$$

should be zero on average. This signal is the TD error and is used as the teaching signal to learn the value function:

$$\Delta V(x(t-1)) = \alpha \delta(t),$$

where α is a learning rate.

The policy is usually defined via the action value function $Q(x(t), a)$, which represents how much future rewards the agent would get by taking the action a at state $x(t)$ and following the current policy in subsequent steps. One common way for stochastic action selection that encourages exploitation is to compute the probability to take an action by the soft-max function:

$$P(a|x(t)) = \frac{e^{\beta Q(x(t), a)}}{\sum_{d=1}^M e^{\beta Q(x(t), d)}},$$

* Corresponding author. Tel.: +81-774-95-1073; fax: +81-774-95-1259.

E-mail address: nicolas@atr.co.jp (N. Schweighofer).

¹ Current address: ATR Human Information Science Laboratories.

where the meta-parameter β is called the inverse temperature.

1.2. The importance of appropriate meta-parameter values

Crucial to successful reinforcement learning is the careful setting of the three meta-parameters α , β and γ .

- The learning rate α is key to maximize the speed of learning, as small learning rates induce slow learning, and large learning rates induce oscillations.
- The inverse temperature β controls the exploitation–exploration trade-off. Ideally, β should initially be low to allow large exploration when the agent does not have a good mapping of which actions will be rewarding, and gradually increase as the agent reaps higher and higher rewards.
- The discount factor γ specifies how far in the future rewards should be taken into account. If γ is small, the agent learns to behave only for short-term rewards. Although setting γ close to 1 promotes the agent to learn to act for long-term rewards, there are several reasons γ should not be set too large. First any real learning agent, either artificial or biological, has a limited lifetime. A discounted value function is equivalent to a non-discounted value function for an agent with a constant death rate of $1 - \gamma$. Second, an agent has to acquire some rewards in time; for instance, an animal must find food before it starves; a robot must recharge its battery before it is exhausted. Third, if the environmental dynamics is highly stochastic or the dynamics is non-stationary, long-term prediction is doomed to be unreliable. Finally, the complexity of learning a value function increases with the increase of $1/(1 - \gamma)$ (Littman et al., 1995).

In many applications, the meta-parameters are hand-tuned by the experimenter, and heuristics are often devised to schedule the value of these meta-parameters as learning progresses. Typically, α is set to decay as the inverse of time, and β to increase linearly with time. However, these algorithms cannot deal with non-stationary environments.

Several robust methods to dynamically adjust meta-parameters have been proposed. We (Schweighofer & Arbib, 1998) proposed a biological implementation of the IDBD (incremental delta bar delta) algorithm (Sutton, 1992) to tune α . The model improves learning performance by automatically setting near optimal synaptic learning rates for each synapse. To direct exploration, Thrun (1992) used a competence network that is trained to estimate the utility of exploration. Ishii et al. (2002) proposed to directly tune β as a function of the inverse of the variance of the action value function. However, although these algorithms are effective in balancing exploration and exploitation, they are specific to each meta-parameters and often make severe computation and memory requirements.

2. Learning meta-parameters with stochastic gradients

Gullapalli (1990) introduced the Stochastic Real Value Units (SRV) algorithm. An SRV unit output is produced by adding to the weighted sum of its input pattern a small perturbation that provides the unit with the variability necessary to explore its activity space. When the action increases the reward, the unit’s weights are adjusted such that the output moves in the direction in which it was perturbed.

Here, we expand the idea of SRV units and take it on to a level higher—that is, we propose that neuromodulator neurons, which project either to the networks that compute the value function, or to the network that selects the actions, are SRV-type units. The ‘gamma neuron’, is governed by

$$\gamma(t) = 1 - \frac{1}{e^{\gamma_b(t)}}$$

where $\gamma_b(t)$ is given by:

$$\gamma_b(t) = \gamma_{b_0} + \sigma_\gamma(t),$$

where γ_{b_0} is a mean activity term, and $\sigma_\gamma(t)$ is a noise term with mean 0, drawn from a Gaussian distribution of mean 0, and variance v . A new value is drawn every n time steps, with $n \gg 1$. Because of this slow change, we assume that this correspond to a small spontaneous change in the tonic firing of the gamma neuron.

To update the mean activity γ_{b_0} , we compute the difference between a short-term and a long-term running average of the reward. Then, the correlation between this difference and the perturbation gives the meta-learning equation

$$\Delta\gamma_{b_0} = \mu(\bar{r}(t) - \bar{\bar{r}}(t))\sigma_\gamma(t),$$

where μ is a learning rate, $\bar{r}(t)$ and $\bar{\bar{r}}(t)$ are, respectively, mid-term and long-term reward running averages,

$$\Delta\bar{r}(t) = \frac{1}{\tau_1}(-\bar{r}(t) + r(t)) \quad \text{and}$$

$$\Delta\bar{\bar{r}}(t) = \frac{1}{\tau_2}(-\bar{\bar{r}}(t) + \bar{r}(t)),$$

where τ_1 and τ_2 are time constants. If a perturbation in the tonic firing rate yields a reward that is better than what the agent is used to, the neuromodulator neuron’s tonic firing rate moves in the direction of the perturbation.

Similar equations govern the behavior of α and β , with the notable difference that because these meta-parameters are not constrained to belong to the interval $[0,1]$, the firing rate is given by the exponential of the mean activity. Thus, the base algorithm is largely independent of the meta-parameter; further, it is independent of the base reinforcement learning method (i.e. Q-learning, Sarsa, actor-critic, etc...).

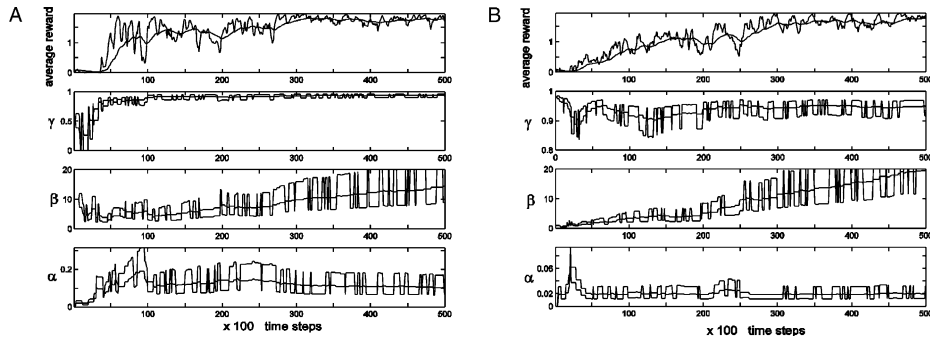


Fig. 1. Evolution of the average reward and of the three meta-parameters for the MDP task with 10 states, with Q-learning as the base reinforcement learning algorithm. For each A and B: Top panel. Short- and long-term average rewards. Bottom three panels: meta-parameters with and without the perturbations. (A) Initial conditions are $\gamma = 0.5$, $\beta = 10$, and $\alpha = 0$. (B) Initial conditions are $\gamma = 0.99$, $\beta = 1$, and $\alpha = 0$. Note the differences in scales.

3. Results

3.1. Discrete space and time

We tested our algorithm on a simulation of a Markov decision problem (MDP) task. In the experiment (Tanaka et al., 2002), on each step, one of four visual stimuli was presented on the screen and the subject was asked to press one of two buttons—one leading to a positive reward, the other to a negative reward (a loss). The subject’s button press determined both the immediate reward and the next stimulus. The state transition and the amount of reward were such that a button with less reward had to be pressed on some steps in order to obtain much larger reward in a subsequent step. In the simulations, there were N states and $2N$ rewards: 1 large reward (R_L), $N - 1$ small rewards (+1 each), $N - 1$ small losses (-1 each), and 1 large loss ($-R_L$). In general, we set $R_L > N$; therefore, the best behavior is to lose $N - 1$ small losses to get one large reward. The worse behavior is to win $N - 1$ small rewards and lose $-R_L$.

We varied the number of states (4, 8, or 10, with respective large rewards: 6, 12, and 15), the initial conditions, and the base reinforcement learning method: Q-learning, Sarsa, and the actor-critic. Further, we varied the number of meta-parameters that could be adjusted (i.e. γ alone, or γ and β , or finally γ , α , and β). In each case, as can be seen for the example shown in Fig. 1, the algorithm did not only find appropriate values of the meta-parameters, but also controlled the time course of these meta-parameters in a dynamic, adaptive manner.

In the simulation corresponding to Fig. 1, we used Q-learning with a 10 states MDP, and the three meta-parameters were learned. Parameters of the simulation were $\tau_1 = 100$, $\tau_2 = 100$, $\mu = 0.2$, $\nu = 1$, and length of the perturbation $n = 100$ time steps. Whether the initial value of γ is initially too small (Fig. 1A) or too large (Fig. 1B), γ converges towards ~ 0.95 . Further for both β and α , there is a dynamic adjustment. If β is set high initially, it first decreases, such that there is large exploration. But as

learning progresses, exploration ‘cost’ a lot of missed rewards to the agent. Thus, β keeps on increasing; and there is only little exploration (Fig. 1A and B). Similarly if α is initially set to a small value close to 0, then it initially increases before slowly converging to a smaller value.

To test the robustness of the algorithm, we varied the number of states and the initial conditions (for γ -only adaptation). We compared the value of γ that gives the highest reward in a non-adaptive algorithm (γ_{fix}), and the asymptotic value of γ given by our meta-learning algorithm. With 4 states, γ_{fix} is around 0.73. Simulations showed that whether the initial value is low (0.5), or high (0.97), γ converges towards γ_{fix} . We find similar behavior in the case of eight states. γ_{fix} is then around 0.92, and whether the initial value is low (0.75), or high (0.97), γ converges towards γ_{fix} .

We then simulated a dynamic environment for an 8-state MDP. After 20,000 time steps, the large reward (R_L) and loss are suddenly changed from 2 and -2 , respectively, to 12 and -12 , respectively. Thus, what was initially a short-term task with at most two-step planning (a small loss once

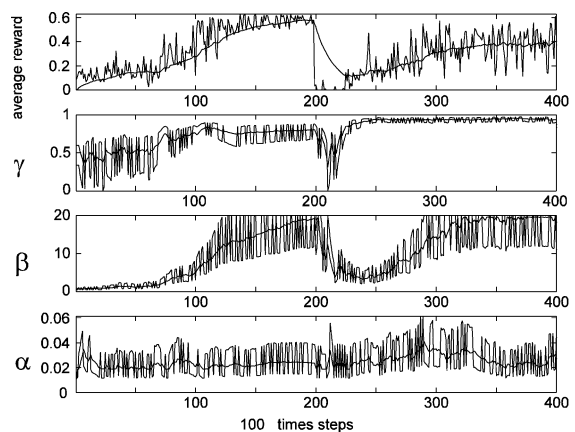


Fig. 2. Adaptation of the meta-parameters to a dynamic environment for an eight state MDP. At 20,000 time steps the large reward and the large loss go from 2 and -2 , respectively, to 12 and -12 , respectively.

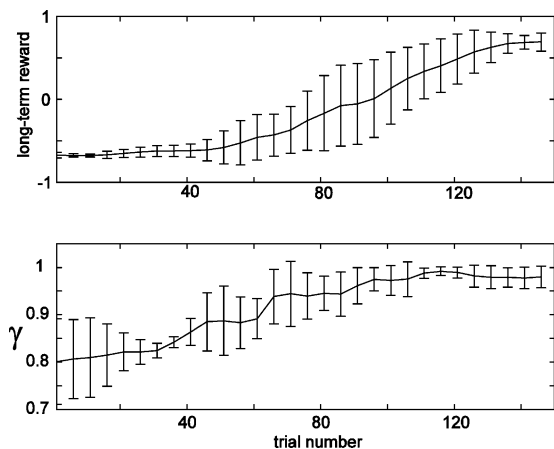


Fig. 3. Evolution of the long-term average reward and the meta-parameter γ in the swing up pendulum task (10 learning runs).

in eight steps), becomes a long-term task in which the agent must lose seven times before winning a large reward. As seen in Fig. 2, the change in reward is immediately followed by a drop to zero in short-term reward, as all the action values are now obsolete. However, the large increase in randomness in action selection (as seen by the drop in β) allows re-learning of a new value function with a larger γ , appropriate for this long-term planning task.

3.2. Continuous space and time

To further validate the robustness of the above meta-learning algorithm, we applied it to the tuning of the discount factor in the control of a pendulum swing-up in which the output torque is limited: the controller has to make preparatory swings to successfully swing up the pendulum. We simulated the continuous-time version of TD learning, which uses the gradient of the learned value function for calculating a greedy policy (Doya, 2000). The reward is the minus of the cosine of the rotation angle.

Meta-learning parameters were $\tau_1 = 100$ s, $\tau_2 = 100$ s, $\mu = 0.025$, $\nu = 0.3$, and duration of the perturbation $n = 400$ s (corresponding to 20 swing up trials). (Please refer to Doya, 2000 for other simulation details).

Fig. 3 shows the adaptation of the discount time scale for 10 runs. Learning was started with $\gamma = 0.8$. As can be seen on the figure, good final performance (as measured by the long-term reward) is attained for $\gamma \sim 0.98$. Note that for the fixed discount factor, $\gamma = 0.8$, the controller can never learn how to swing to maintain the pendulum in the upward position.

4. Discussion

We proposed a simple, yet robust and generic, algorithm that does not only finds near-optimal meta-parameters, but also controls the time course of the meta-parameters in a dynamic, adaptive manner.

Since we introduced five extra parameters, i.e. two global parameters, τ_1 , τ_2 , and three parameters specific to each neuromodulator neuron, μ , ν , and n , do we need meta-meta-learning algorithms, or even higher order learning mechanisms, to tune these parameters? Let's notice that we have actually only two free meta-meta-parameters T and A (for time and amplitude). First, the time constants of the reward averaging are of the same order of magnitude, and are obviously related to the duration of the perturbation. Second, the learning rate and the perturbation size are also related (see the meta-learning equation). We ran extensive simulations to study the effect of τ_1 , τ_2 upon final performance (MDP task, 10 states, Q-learning), and found good asymptotic performance for $5000 > \tau_1 > 5$ and $5000 > \tau_2 > 5$. Thus, as the algorithm is extremely robust regarding the T meta-meta-parameter, we can assume that its value is genetically determined, and perhaps related to the wake-sleep cycle of the animal (see below). As the value of the A parameter is more sensitive, it is not impossible that a meta-meta-learning algorithm operates to tune it. The robust and biologically plausible algorithm that we earlier proposed (Schweighofer & Arbib, 1998) for the setting of learning rates, and whose performance is almost independent of the choice of the meta-parameters (within very large ranges at least) could be possibly used.

Because, our algorithm makes only minimal computational and memory requirements, and does not depend on the exact underlying reinforcement methods, we suggest that it can be used in biological meta-learning (Fig. 4). Doya (2002) proposed a theory of the roles of neuromodulators in terms of setting the meta-parameters. Specifically in this theory, serotonin controls the discount factor γ ; norepinephrine controls the inverse temperature β , and acetylcholine controls the learning rate α . Our meta-learning algorithm makes the three following predictions.

1. Spontaneous fluctuations of the tonic firing of the neuromodulator neuron. We suggest that the perturbations may arise naturally with the wake-sleep cycle

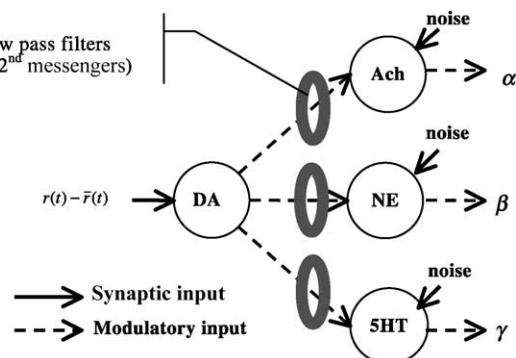


Fig. 4. Possible biological implementation of the meta-learning algorithm. DA: dopamine. Ach. Acetylcholine. NE. Norepinephrine. 5HT. Serotonin.

and/or the level of activity of the animal. This type of modulation of is well known for the serotonergic and noradrenergic neurons (Aston-Jones & Bloom, 1981; Jacobs & Fornal, 1993).

2. Mid-term and long-term rewards averages. In their model of motivation and emotion, Solomon and Corbit (Solomon and Corbit 1974) used a difference between a mid-term running average of rewards and a long-term running average of punishments to explain the negative affect that follows positive affects (or vice versa). It was recently proposed that the dopamine phasic firing carries the short-term reward and the dopamine tonic firing carries a long-term opponent signal (Daw et al., 2002). If we consider that the long-term opponent process is simply the opposite of the longer-term reward process, then the dopamine neurons could carry the signal performing the subtraction of short- and long-term rewards averages. The signal $r(t) - \bar{r}(t)$ would then be carried by dopamine to the other neuromodulator neurons and (linear) filtering processes in these neurons, such as second messengers or diffusion, would then transform this signal into $\bar{r}(t) - \bar{\bar{r}}(t)$, the signal required for our meta-learning rule.
3. Dopamine-dependent plasticity of neuromodulator neurons (see meta-learning equation). Dopamine neurons project to serotonergic, noradrenergic, and cholinergic neurons (Maeda et al., 1991; Day & Fibiger, 1993; Haj-Dahmane, 2001). We predict that dopamine-dependent plasticity exists in these neurons. Thus, we suggest that the phasic dopamine signal is the reward needed for reinforcement learning and the complete dopamine signal is the reward for meta-learning of reinforcement learning.

Acknowledgements

We thank Etienne Burdet, Stephan Schaal, and Fredrik Bissmarck for comments on a first draft. This work was supported by CREST, JST.

References

- Aston-Jones, G., & Bloom, F. E. (1981). Activity of norepinephrine-containing locus coeruleus neurons in behaving rats anticipates fluctuations in the sleep–waking cycle. *Journal of Neuroscience*, *1*(8), 876–886.
- Daw, N. D., Kakade, S., & Dayan, P. (2002). Opponent interactions between serotonin and dopamine. *Neural Networks*, *15*, 603–616.
- Day, J., & Fibiger, H. C. (1993). Dopaminergic regulation of cortical acetylcholine release: effects of dopamine receptor agonists. *Neuroscience*, *54*(3), 643–648.
- Doya, K. (2000). Reinforcement learning in continuous time and space. *Neural Computations*, *12*(1), 219–245.
- Doya, K. (2002). Metalearning and neuromodulation. *Neural Networks*, *15*, 495–506.
- Gullapalli, V. (1990). A stochastic reinforcement learning algorithm for learning real-valued functions. *Neural Networks*, *3*, 671–692.
- Haj-Dahmane, S. (2001). D2-like dopamine receptor activation excites rat dorsal raphe 5-HT neurons in vitro. *European Journal of Neuroscience*, *14*(1), 125–134.
- Ishii, S., Yoshida, W., & Yoshimoto, J. (2002). Control of exploitation–exploration meta-parameters in reinforcement learning. *Neural Networks*, *15*, 665–687.
- Jacobs, B. L., & Fornal, C. A. (1993). 5-HT and motor control: a hypothesis. *Trends in Neuroscience*, *16*(9), 346–352.
- Littman, M. L., Dean, T. L., et al (1995). On the complexity of solving Markov decision problems. Eleventh International Conference on Uncertainty in Artificial Intelligence.
- Maeda, T., Kojima, Y., Arai, R., Fujimiya, M., Kimura, H., Kitahama, A., & Geffard, M. (1991). Monoaminergic interaction in the central nervous system: a morphological analysis in the locus coeruleus of the rat. *Comparative Biochemistry and Physiology C*, *98*(1), 193–202.
- Schultz, W. (1998). Predictive reward signal of dopamine neurons. *Journal of Neurophysiology*, *80*(1), 1–27.
- Schweighofer, N., & Arbib, M. A. (1998). A model of cerebellar metaplasticity. *Learning Memory*, *4*(4), 421–428.
- Solomon, R. L., & Corbit, J. D. (1974). An opponent process theory of motivation. I. Temporal dynamics of affect. *Psychological Review*, *81*, 119–145.
- Sutton, R (1992). Adapting bias by gradient descent: an incremental version of the delta-bar-delta. *Tenth National Conference on Artificial Intelligence*. Cambridge, MA: MIT Press.
- Tanaka, S., Doya, K., Okada, G., Ueda, K., Okamoto, Y., & Yamawaki, S. (2002). Functional MRI study of short-term and long-term prediction of reward. *Proceedings of the Eighth International Conference on Functional Mapping of the Human Brain, Sendai, Japan*: 1062.
- Thrun, S. B. (1992). In D. A. White, & D. A. Dofge (Eds.), *The role of exploration in learning control. Handbook of intelligent control: Neural, fuzzy, and adaptive approaches*. Florence, Kentucky: Van Nostrand.