Self-Adjusting Reinforcement Learning

Ralf Der * and Michael Herrmann **

* Universität Leipzig, Inst. f. Informatik, Postfach 920, 04109 Leipzig, Germany
** RIKEN, Lab. f. Information Representation, 2-1 Hirosawa, Wako 351-01, Saitama, Japan

Abstract

We present a variant of the $\mathcal{Q}$-learning algorithm with automatic control of the exploration rate by a competition scheme. The theoretical approach is accompanied by systematic simulations of a chaos control task. Finally, we give interpretations of the algorithm in the context of computational ecology and neural networks.

I. Introduction

Reinforcement learning [3, 4], originally a paradigm in psychological learning theory, has established itself during the last decade as a class of powerful algorithms in nonlinear control. Several algorithms are available that try to approximate a value function over a set of system states and possible state transitions. Given an initial state, this function can be used to choose a sequence of state transitions that approaches a state of maximal value. In order to approximate the value function the state space is explored by random transitions with an externally increased bias for 'good' transitions.

We present a novel approach to reinforcement learning in which state transitions compete for reinforcement rather than undergo a controlled adaptation. In this way the algorithm becomes robust against parameter changes and is, hence, expected to be more reliable in complex control tasks.

In the next section we review the $\mathcal{Q}$-learning scheme [4, 5] which has some formal similarities with the present algorithm and which is therefore well suited for a comparison (cf. section V.). Our approach is based on a competition among state transitions which is locally governed by the Fisher-Eigen-equations [2], described in section III. Global properties of self-adjusting reinforcement learning are discussed in section IV. The results of numerical simulations are presented in section V. Most interestingly, the novel learning algorithm allows for challenging interpretations in a variety of contexts, cf. section VI.

II. $\mathcal{Q}$-learning

In $\mathcal{Q}$-learning the value function assings a $\mathcal{Q}$uality measure to each pair (i,a), where i denotes a system state and a a control action. $\mathcal{Q}(i,a)$ is adapted to predict the *total discounted future reinforcement* when performing first action a and following the currently best possible strategy given by $a = \operatorname{argmax}_a \mathcal{Q}(i,a)$ thereafter. Discounting means that a reinforcement signal r arriving t time steps later is considered to have a value decreased by a factor γ^t, where $\gamma \leq 1$ is given as the time horizon of the system. In addition to the present state which is in fact only a label, the only knowledge the learning algorithm receives about the system is the reinforcement signal. r is state-dependent and assumes positive values for a goal state, negative values for failure state and is zero otherwise. Other choices of r may be useful in accordance to context. $\mathcal{Q}(i,a)$ is updated by

$$\Delta\mathcal{Q}(i,a) = \epsilon\,(r(i) + \gamma V(i \oplus a) - \mathcal{Q}(i,a)) \quad (1)$$

$$V(i) = \max_a \mathcal{Q}(i,a) \quad (2)$$

$$a(t) = a \text{ with probability } p_{i,a}, \quad (3)$$

where $p_{i,a}$ is maximal for $\mathcal{Q}(i,a)$ being maximal w.r.t. a. A common choice is

$$p_{i,a} = \frac{e^{\mathcal{Q}(i,a)/\beta}}{\sum_b e^{\mathcal{Q}(i,b)/\beta}}, \quad (4)$$

where for a small exploration rate β the currently best action is strongly favored. We write $j = i \oplus a$ if the system moves deterministically to state j if the action a is applied in state i. For systems which are intrinsically stochastic apart from the random selection of actions the result of the operation $i \oplus a$ is state j with probability p_{ij}^a, where $\sum_j p_{ij}^a = 1, \ \forall\, i,a$. In order to avoid confusion of p_{ij}^a and the probability $p_{i,a}$ of choosing action a in state i we will consider in the following only intrinsically deterministic systems.

$\mathcal{Q}$-learning has been proven [6] to find optimal strategies in Markovian systems when each state is

visited potentially infinitely often and the adaptation rate ϵ satisfies $\sum \epsilon(t) = \infty$ and $\sum \epsilon(t)^2 < \infty$. For large state spaces and many actions, however, the finite computing time performance strongly depends on the time course of ϵ and particularly on the exploration strategy given in terms of $p_{i,a}$, which accounts for the avoidance of local minima. $\mathcal{Q}$-learning requires to fix the time course of the variables ϵ and β. Little is known about optimal cooling schemes for β.

III. Self-adjusting quasispecies

The present algorithm is based upon a population dynamics inspired by the approach described in [2]. Considering a *fixed* state i the probabilities $p_{i,a}$ are interpreted as relative frequencies p_a (omitting index i) of a 'species' a which has fitness V_a. The frequency of the species evolves according to

$$\tau_p \Delta p_a = (V_a - \langle V \rangle)\, p_a, \tag{5}$$

where

$$\langle V \rangle = \sum_a p_a V_a \tag{6}$$

is the average fitness of the individuals living at site i. Hence, p_a grows if the fitness of a exceeds the average fitness and decreases otherwise. Eq. (5) is a discrete version the Fisher-Eigen equations of prebiotic evolution [2], which have the following properties. The probabilities are conserved, i.e.

$$\sum_a p_a(t) = \text{ constant } \quad \forall t. \tag{7}$$

The Fisher-Eigen equations are explicitly solvable:

$$p_a(t) = \frac{e^{V_a t} p_a(0)}{\sum_b e^{V_b t} p_b(0)}, \tag{8}$$

such that in the limit $t \to \infty$

$$p_a(t) = \begin{cases} 1 & \text{for } a = \max_a V_a \\ 0 & \text{otherwise} \end{cases}. \tag{9}$$

Further, there is a Lyapunov function

$$L = \sum_a p_a V_a \tag{10}$$

which satisfies

$$\frac{d}{dt} L \geq 0. \tag{11}$$

The relevance of this schemes becomes clear is the fitness values are not fixed quantities rather than being determined by the value of a subsequent state in the reinforcement dynamics, namely:

$$V_a \equiv r(i) + \gamma V(i \oplus a) \tag{12}$$

However, the assumption of constant fitness implicit in the Fisher-Eigen equations does not hold in this case. In place of the evolution of a single population we have now the situation of co-evolving subpopulation residing at sites i, where the subpopulations interact by backward transmission of discounted reinforcement and forward activation by choosing a control action based on the current $p_{i,a}$ values.

IV. Self-adjusting reinforcement learning

Returning to the reinforcement learning scheme Eq. (5) becomes

$$\tau_P \Delta p_{i,a} = r(i) + \gamma V(i \oplus a) - \langle r(i) + \gamma V(i \oplus a) \rangle. \tag{13}$$

The multiplication by p_a in Eq. (5) is now hidden in the choice of a according to the probabilities $p_{i,a}$, i.e. (13) is a stochastic approximation of the coupled Fisher-Eigen equations. The average in (13) can be rewritten as

$$V(i) \stackrel{\text{def}}{=} \langle r(i) + \gamma V(i \oplus a) \rangle = r(i) + \gamma \sum_a p_{i,a} V(i \oplus a), \tag{14}$$

which is the expected reinforcement at state i when following the strategy with stochastic action choice for subsequent time steps rather than the currently best possible ('greedy') strategy as in $\mathcal{Q}$-learning. When formulating also Eq. (14) as a stochastic approximation scheme in order to avoid performing the explicit sum in each time step, we obtain an update rule for V which together with Eq. (13) forms the main equation of on-line version of the present algorithm.

$$\tau_p \Delta p_{i,a} = r(i) + \gamma V(i \oplus a) - V(i) \tag{15}$$

$$\tau_V \Delta V(i) = r(i) + \gamma V(i \oplus a) - V(i) \tag{16}$$

(15) and (16) have to be solved simultaneously which is numerically convenient because of identical r.h.s.'s. Thus, the self-adjusting reinforcement learning algorithm depends on the choice of the *fixed* times scales τ_p and τ_V in contrast to cooling schemes in $\mathcal{Q}$-learning. In order to analyze the complementary equations (15) and (16) we consider the averaged versions (13) and (14).

If the changes in the V values are neglectable compared to the time scale in (13) we recover the situation of the discrete Fisher-Eigen equation (5). The fitness is constant on short time scales such that the convergence and normalization properties are preserved. Since, however, for $\tau_p \ll \tau_V$ the probabilities converge to zero or one further exploration becomes impossible and the resulting strategy remains suboptimal.

If on the other hand $\tau_p \gg \tau_V$ we can look separately for quasi-stationary solutions of (16). We introduce a matrix

$$M_{ij} = \begin{cases} p_{i,a} & \text{if } j = i \oplus a \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

for the intrinsically deterministic case or analogously for the intrinsically stochastic case

$$M_{ij} = \sum_a p_{ij}^a p_{i,a} \tag{18}$$

and write the stationary state of Eq. (14) in vector notation

$$\mathbf{V} = \mathbf{r} + \gamma \mathbf{M}\mathbf{V}. \tag{19}$$

Since $\mathbf{M}$ is a probability matrix all of its eigenvalues λ obey $|\lambda| \leq 1$. Hence, it is possible to solve (19) for $0 < \gamma < 1$ and the solution of (19) is stable.

$$\mathbf{V} = (\mathbf{1} - \gamma \mathbf{M})^{-1} \mathbf{r} \tag{20}$$

The current value function given by the solution (20) can be used in Eq. (15) adapting on a slower time scale.

Since most of the p-values decay exponentially necessary changes of the $p_{i,a}$ due to changes in the V will by difficult to maintain since action with $p_{i,a}$ close to zero are infrequently explored. In order to avoid the convergence to such local minima the decay of the probabilities should by restricted to small positive limit values $\eta > 0$. In this way the convergence rate is improved and effects of small numbers are removed. Theoretically such a restriction is, however, not necessary since the probabilities also recover exponentially fast from small values. For comparable time scales τ_p and τ_V Eq. (14) ensures the conservation of probability for the continuous version of (13). In the case of a discretization or a stochastic approximation, however, the probabilities have to be normalized explicitly.

V. Simulation results

We have applied the proposed learning scheme to a pedagogical centering task, where an analytical solution both for $\mathcal{Q}$-learning as well as for self-adjusting reinforcement learning is obtainable and in coincidence with the numerical results. Other successful simulations are the cart-pole problem and the control of unstable periodic orbits in a Mackey-Glass system.

Here we will present a more systematic numerical study on a simple chaotic stabilization task, namely the stabilization of an unstable fixed point in the logistic map, cf. [1]. The algorithm runs for a fixed number of 100000 time steps using inputs from a partition into 200 categories of the one-dimensional state space and a reinforcement signal which assumes non-zero values whenever the state passes near the fixed point. When testing possible combinations of *fixed* τ_p and τ_V, the latter parameter turned out not to be critical. Even values different by several orders of magnitude did not change the performance of the algorithm. In contrast, when τ_p is too large the control task cannot be solved in limited time. For a smaller time scale τ_p on the other hand the convergence of the p-values is too quick such that only a poor solution is reached before sufficiently exploring the state-action space was possible. Fig. 1 indicates the regions of an average control time of less than 10 per cent above the optimal stabilization time.

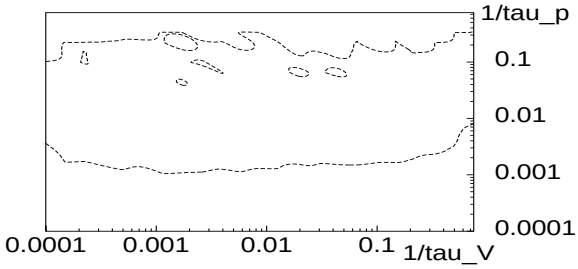

Figure 1: Region in the (τ_p, τ_V) parameter space which allows for an average stabilization time from a random initial state that is less than 10 per cent above the minimal average time in a chaos control task.

For comparison the $\mathcal{Q}$-learning algorithm has been applied to the same problem. The learning rate ϵ and the exploration rate β have a similar meaning as τ_V^{-1} and τ_p^{-1} in the self-adjusting scheme. For fixed values of ϵ and β the parameter space region which allows for a solution of the control task is a very small area at large ϵ and relatively small β. If β decays linearly to zero during the learning period and ϵ decays algebraically a plot similar to Fig. 1 can be made for the starting values of ϵ and β. In this case the successful region is qualitatively similar, although smaller.

VI. Discussion

The presented learning algorithm allows for challenging interpretations in the contexts of computational ecology and neural dynamics. The main field for applications of reinforcement learning algorithms is, however, control. Therefore we have been mainly using this language and referred to the i as system states and to the a as control actions.

Computational ecology: Self-adjusting reinforcement learning has a background in computational ecology. In order to illustrate this relation we will give now a more complete interpretation of the learning algorithm on a social model.

We consider a population of traders living in a city i in which different kinds a of traders have specific trade relations to certain other other cities $j = i \oplus a$. In any city the goods offered for sale are either produced there (r) or bought from other cities. The demand $V(i)$ for goods from city i depends, thus, from $r(i)$ as well as from the value of goods brought from elsewhere to i discounted by a factor γ for the cost of transportation. If a trade mission to city $j = i \oplus a$ has turned out

to be more successful than usual that trade relation will be more frequently exploited in future (update p), also the trade coordination department of i should announce the success (update V) in order to increase the demand. In a system as simple as described here the produced goods should arrive as quickly as possible at the consumers which are the leaves in the evolving tree-like trade structure.

Neural dynamics: In the context of neural systems the action probabilities $p_{i,a}$ are the efficacies of synaptic connections between neurons i and $j = i \oplus a$. V_i is the mean firing rate of neuron i. If an action potential is send from i those efficacies increase which relate to activated neurons j. The activation is more likely if the connection $p_{i,a}$ is strong. Hence, the dynamics of the neural implementation follows directly Eqs. (13) and (14) rather than their stochastic counterparts. The learning rule for the synapses would lead to single-output neurons in contrast to the network structure in real neural systems. However, by requiring a minimal number of action potentials arriving at a neuron to be activated and introducing mechanisms to keep the total activity constant more complex connectivity structures arise. The resulting neural arrangement can detect coincidences of arriving spikes and is functionally similar to a synfire chain architecture.

VII. Conclusion

We have presented a variant of the $\mathcal{Q}$-learning algorithm with automatic control of the exploration rate by a competition scheme similar to the Fisher-Eigen-equations known from evolutionary dynamics. The self-adjusting reinforcement learning algorithm is different from $\mathcal{Q}$-learning in that the state-action value function is replaced by the evolution of action probabilities for each state. In addition, the adjustable variables are determined by an average over possible actions at any later time weighted by the adaptive transition probabilities rather than the currently optimal strategy. The algorithm is simpler than $\mathcal{Q}$-learning insofar as no parameter cooling schemes are necessary. In particular, the self-adjustment of the exploration rate is superior to a fixed scheme when the reinforcement signals are changing in time. The implementation of its off-line version as a reinforcement learning neural network will be studied in a forthcoming paper.

Acknowledgement

One of the authors (RD) gratefully acknowledges the hospitality of the Frontier Research Program at RIKEN (Tokyo).

References

[1] R. Der, M. Herrmann (1994) $\mathcal{Q}$-learning chaos controller. Proc. WCNN'94, Orlando, Florida, p. 2472-2475.

[2] M. Eigen, J McCaskill, P. Schuster (1989) The molecular quasispecies. *Adv. Chem. Phys.* **75**, 149-263.

[3] R. S. Sutton (1988) Learning to predict by the method of temporal differences. *Machine Learning* **3**, 9-44.

[4] C. J. C. H. Watkins (1989) *Learning from delayed rewards.* PhD Thesis, Cambridge Univ., Cambridge, England.

[5] C. J. C. H. Watkins (1992) $\mathcal{Q}$-learning. *Machine Learning* **8**, p. 279-292.

[6] C. J. C. H. Watkins, P. Dayan (1992) Technical note: $\mathcal{Q}$-learning. *Machine Learning* **8**, 279-292.

E-mail: der@informatik.uni-leipzig.de,
michael@zoo.riken.go.jp, mherrma@gwdg.de
New address of MH: MPI SF, Postfach 2853,
37018 Göttingen, Germany