Technical Note: Q-Learning

Article *in* Machine Learning · May 1992

DOI: 10.1007/BF00992698 · Source: DBLP

2 authors, including:

Christopher Watkins
Royal Holloway, University of London
45 PUBLICATIONS **18,544** CITATIONS

Machine Learning, 8, 279-292 (1992)

Technical Note
Q-Learning

CHRISTOPHER J.C.H. WATKINS
25b Framfield Road, Highbury, London N5 1UU, England

PETER DAYAN
Centre for Cognitive Science, University of Edinburgh, 2 Buccleuch Place, Edinburgh EH8 9EH, Scotland

Abstract. Q-learning (Watkins, 1989) is a simple way for agents to learn how to act optimally in controlled Markovian domains. It amounts to an incremental method for dynamic programming which imposes limited computational demands. It works by successively improving its evaluations of the quality of particular actions at particular states.

This paper presents and proves in detail a convergence theorem for Q-learning based on that outlined in Watkins (1989). We show that Q-learning converges to the optimum action-values with probability 1 so long as all actions are repeatedly sampled in all states and the action-values are represented discretely. We also sketch extensions to the cases of non-discounted, but absorbing, Markov environments, and where many Q values can be changed each iteration, rather than just one.

Keywords. Q-learning, reinforcement learning, temporal differences, asynchronous dynamic programming

1. Introduction

Q-learning (Watkins, 1989) is a form of model-free reinforcement learning. It can also be viewed as a method of asynchronous dynamic programming (DP). It provides agents with the capability of learning to act optimally in Markovian domains by experiencing the consequences of actions, without requiring them to build maps of the domains.

Learning proceeds similarly to Sutton's (1984; 1988) method of temporal differences (TD): an agent tries an action at a particular state, and evaluates its consequences in terms of the immediate reward or penalty it receives *and* its estimate of the value of the state to which it is taken. By trying all actions in all states repeatedly, it learns which are best overall, judged by long-term discounted reward. Q-learning is a primitive (Watkins, 1989) form of learning, but, as such, it can operate as the basis of far more sophisticated devices. Examples of its use include Barto and Singh (1990), Sutton (1990), Chapman and Kaelbling (1991), Mahadevan and Connell (1991), and Lin (1992), who developed it independently. There are also various industrial applications.

This paper presents the proof outlined by Watkins (1989) that Q-learning converges. Section 2 describes the problem, the method, and the notation, section 3 gives an overview of the proof, and section 4 discusses two extensions. Formal details are left as far as possible to the appendix. Watkins (1989) should be consulted for a more extensive discussion of Q-learning, including its relationship with dynamic programming and TD. See also Werbos (1977).

2. The task for $\mathcal{Q}$-learning

Consider a computational agent moving around some discrete, finite world, choosing one from a finite collection of actions at every time step. The world constitutes a controlled Markov process with the agent as a controller. At step n, the agent is equipped to register the state x_n ($\in X$) of the world, an can choose its action a_n ($\in \mathcal{A}$)[1] accordingly. The agent receives a probabilistic reward r_n, whose mean value $\mathcal{R}_{x_n}(a_n)$ depends only on the state and action, and the state of the world changes probabilistically to y_n according to the law:

$$\text{Prob}\,[y_n = y | x_n, a_n] = P_{x_n y}[a_n].$$

The task facing the agent is that of determining an optimal policy, one that maximizes total discounted expected reward. By discounted reward, we mean that rewards received s steps hence are worth less than rewards received now, by a factor of γ^s ($0 < \gamma < 1$). Under a policy π, the value of state x is

$$V^\pi(x) \equiv \mathcal{R}_x(\pi(x)) + \gamma \sum_y P_{xy}[\pi(x)] V^\pi(y$$

because the agent expects to receive $\mathcal{R}_x(\pi(x))$ immediately for performing the action π recommends, and then moves to a state that is 'worth' $V^\pi(y)$ to it, with probability $P_{xy}[\pi(x)]$. The theory of DP (Bellman & Dreyfus, 1962; Ross, 1983) assures us that there is at least one optimal stationary policy π^* which is such that

$$V^*(x) \equiv V^{\pi^*}(x) = \max_a \left\{ \mathcal{R}_x(a) + \gamma \sum_y P_{xy}[a] V^{\pi^*}(y) \right\}$$

is as well as an agent can do from state x. Although this might look circular, it is actually well defined, and DP provides a number of methods for calculating V^* and one π^*, assuming that $\mathcal{R}_x(a)$ and $P_{xy}[a]$ are known. The task facing a $\mathcal{Q}$ learner is that of determining a π^* without initially knowing these values. There are traditional methods (e.g., Sato, Abe & Takeda, 1988) for learning $\mathcal{R}_x(a)$ and $P_{xy}[a]$ while concurrently performing DP, but any assumption of certainty equivalence, i.e., calculating actions as if the current model were accurate, costs dearly in the early stages of learning (Barto & Singh, 1990). Watkins (1989) classes $\mathcal{Q}$-learning as incremental dynamic programming, because of the step-by-step manner in which it determines the optimal policy.

For a policy π, define $\mathcal{Q}$ values (or action-values) as:

$$\mathcal{Q}^\pi(x, a) = \mathcal{R}_x(a) + \gamma \sum_y P_{xy}[\pi(x)] V^\pi(y).$$

In other words, the $\mathcal{Q}$ value is the expected discounted reward for executing action a at state x and following policy π thereafter. The object in $\mathcal{Q}$-learning is to estimate the $\mathcal{Q}$

values for an optimal policy. For convenience, define these as $Q^*(x, a) \equiv Q^{\pi^*}(x, a), \forall x, a$. It is straightforward to show that $V^*(x) = \max_a Q^*(x, a)$ and that if a^* is an action at which the maximum is attained, then an optimal policy can be formed as $\pi^*(x) \equiv a^*$. Herein lies the utility of the Q values—if an agent can learn them, it can easily decide what it is optimal to do. Although there may be more than one optimal policy or a^*, the Q^* values are unique.

In Q-learning, the agent's experience consists of a sequence of distinct stages or *episodes*. In the n^{th} episode, the agent:

- observes its current state x_n,
- selects and performs an action a_n,
- observes the subsequent state y_n,
- receives an immediate payoff r_n, and
- adjusts its Q_{n-1} values using a learning factor α_n, according to:

$$Q_n(x, a) = \begin{cases} (1 - \alpha_n) Q_{n-1}(x, a) + \alpha_n [r_n + \gamma V_{n-1}(y_n)] & \text{if } x = x_n \text{ and } a = a_n, \\ Q_{n-1}(x, a) & \text{otherwise,} \end{cases} \tag{1}$$

where

$$V_{n-1}(y) \equiv \max_b \{Q_{n-1}(y, b)\} \tag{2}$$

is the best the agent thinks it can do from state y. Of course, in the early stages of learning, the Q values may not accurately reflect the policy they implicitly define (the maximizing actions in equation 2). The initial Q values, $Q_0(x, a)$, for all states and actions are assumed given.

Note that this description assumes a look-up table representation for the $Q_n(x, a)$. Watkins (1989) shows that Q-learning may not converge correctly for other representations.

The most important condition implicit in the convergence theorem given below is that the sequence of episodes that forms the basis of learning must include an infinite number of episodes for each starting state and action. This may be considered a strong condition on the way states and actions are selected—however, under the stochastic conditions of the theorem, no method could be guaranteed to find an optimal policy under weaker conditions. Note, however, that the episodes need not form a continuous sequence—that is the y of one episode need not be the x of the next episode.

The following theorem defines a set of conditions under which $Q_n(x, a) \to Q^*(x, a)$ as $n \to \infty$. Define $n^i(x, a)$ as the index of the i^{th} time that action a is tried in state x.

Theorem

Given bounded rewards $|r_n| \leq \mathcal{R}$, learning rates $0 \leq \alpha_n < 1$, and

$$\sum_{i=1}^{\infty} \alpha_n{}^i{}_{(x,a)} = \infty, \quad \sum_{i=1}^{\infty} [\alpha_n{}^i{}_{(x,a)}]^2 < \infty, \quad \forall x, a, \tag{3}$$

then $\mathcal{Q}_n(x, a) \to \mathcal{Q}^*(x, a)$ as $n \to \infty$, $\forall x, a$, with probability 1.

3. The convergence proof

The key to the convergence proof is an artificial controlled Markov process called the *action-replay process* ARP, which is constructed from the episode sequence and the learning rate sequence α_n.

A formal description of the ARP is given in the appendix, but the easiest way to think of it is in terms of a card game. Imagine each episode $\langle x_t, a_t, y_t, r_t, \alpha_t \rangle$ written on a card. All the cards together form an infinite deck, with the first episode-card next-to-bottom and stretching infinitely upwards, in order. The bottom card (numbered 0) has written on it the agent's initial values $\mathcal{Q}_0(x, a)$ for all pairs of x and a. A state of the ARP, $\langle x, n \rangle$, consists of a card number (or *level*) n, together with a state x from the real process. The actions permitted in the ARP are the same as those permitted in the real process.

The next state of the ARP, given current state $\langle x, n \rangle$ and action a, is determined as follows. First, all the cards for episodes later than n are eliminated, leaving just a finite deck. Cards are then removed one at a time from top of this deck and examined until one is found whose starting state and action match x and a, say at episode t. Then a biased coin is flipped, with probability α_t of coming out heads, and $1 - \alpha_t$ of tails. If the coin turns up heads, the episode recorded on this card is replayed, a process described below; if the coin turns up tails, this card too is thrown away and the search continues for another card matching x and a. If the bottom card is reached, the game stops in a special, absorbing, state, and just provides the reward written on this card for x, a, namely $\mathcal{Q}_0(x, a)$.

Replaying the episode on card t consists of emitting the reward, r_t, written on the card, and then moving to the next state $\langle y_t, t - 1 \rangle$ in the ARP, where y_t is the state to which the real process went on that episode. Card t itself is thrown away. The next state transition of the ARP will be taken based on just the remaining deck.

The above completely specifies how state transitions and rewards are determined in the ARP. Define $P^{\text{ARP}}_{\langle x,n \rangle, \langle y,m \rangle}[a]$ and $\mathcal{R}_x^{(n)}(a)$ as the transition-probability matrices and expected rewards of the ARP. Also define:

$$P_{xy}^{(n)}[a] = \sum_{m=1}^{n-1} P^{\text{ARP}}_{\langle x,n \rangle, \langle y,m \rangle}[a]$$

as the probabilities that, for each x, n and a, executing action a at state $\langle x,n \rangle$ in the ARP leads to state y of the real process at some lower level in the deck.

As defined above, the ARP is as much a controlled Markov process as is the real process. One can therefore consider sequences of states and controls, and also optimal discounted $\mathcal{Q}^*$ values for the ARP.[2] Note that during such a sequence, episode cards are only removed from the deck, and are never replaced. Therefore, after a finite number of actions, the bottom card will always be reached.

3.1. Lemmas

Two lemmas form the heart of the proof. One shows that, effectively by construction, the optimal $\mathcal{Q}$ value for ARP state $\langle x, n \rangle$ and action a is just $\mathcal{Q}_n(x, a)$. The next shows that for almost all possible decks, $P_{xy}^{(n)}[a]$ converge to $P_{xy}[a]$ and $\mathcal{R}_x^{(n)}(a)$ converge to $\mathcal{R}_x(a)$ as $n \to \infty$. Informal statements of the lemmas and outlines of their proofs are given below; consult the appendix for the formal statements.

Lemma A

$\mathcal{Q}_n(x, a)$ are the optimal action values for ARP states $\langle x, n \rangle$ and ARP actions a.

The ARP was directly constructed to have this property. The proof proceeds by backwards induction, following the ARP down through the stack of past episodes.

Lemma B

Lemma B concerns the convergence of the ARP to the real process. The first two steps are preparatory; the next two specify the form of the convergence and provide foundations for proving that it occurs.

B.1

Consider a discounted, bounded-reward, finite Markov process. From any starting state x, the difference between the value of that state under the finite sequence of s actions and its value under that same sequence followed by any other actions tends to 0 as $s \to \infty$.

This follows from the presence of the discount factor which weighs the $(s + 1)^{\text{th}}$ state by $\gamma^s \to 0$ as $s \to \infty$.

B.2

Given any level l, there exists another yet higher level, h, such that the probability can be made arbitrarily small of straying below l after taking s actions in the ARP, starting from above h.

The probability, starting at level h of the ARP of straying below any fixed level l tends to 0 as $h \to \infty$. Therefore there is some sufficiently high level for which s actions can be safely accommodated, with an arbitrarily high probability of leaving the ARP above l.

B.3

With probability 1, the probabilities $P_{xy}^{(n)}[a]$ and expected rewards $\mathcal{R}_x^{(n)}(a)$ in the ARP converge and tend to the transition matrices and expected rewards in the real process as the level n increases to infinity. This, together with B.2, makes it appropriate to consider $P_{xy}^{(n)}[a]$ rather than the ARP transition matrices $P^{\text{ARP}}_{\langle x,n\rangle,\langle y,m\rangle}[a]$, i.e., essentially ignoring the level at which the ARP enters state y.

The ARP effectively estimates the mean rewards and transitions of the real process over all the episodes. Since its raw data are unbiased, the conditions on the sums and sums of squares of the learning rates $\alpha_{n^i(x,a)}$ ensure the convergence with probability one.

B.4

Consider executing a series of s actions in the ARP and in the real process. If the probabilities $P_{xy}^{(n)}[a]$ and expected rewards $\mathcal{R}_x^{(n)}(a)$ at appropriate levels of the ARP for each of the actions, are close to $P_{xy}[a]$ and $\mathcal{R}_x(a)$, $\forall a, x, y$, respectively, then the value of the series of actions in the ARP will be close to its value in the real process.

The discrepancy in the action values over a finite number s of actions between the values of two approximately equal Markov processes grows at most quadratically with s. So, if the transition probabilities and rewards are close, then the values of the actions must be close too.

3.2. The theorem

Putting these together, the ARP tends towards the real process, and so its optimal $\mathcal{Q}$ values do too. But $\mathcal{Q}_n(a, x)$ are the optimal $\mathcal{Q}$ values for the n^{th} level of the ARP (by Lemma A), and so tend to $\mathcal{Q}^*(x, a)$.

Assume, without loss of generality, that $\mathcal{Q}_0(x, a) < \mathcal{R}/(1 - \gamma)$ and that $\mathcal{R} \geq 1$.

Given $\epsilon > 0$, choose s such that

$$\gamma^s \frac{\mathcal{R}}{1 - \gamma} < \frac{\epsilon}{6}.$$

By B.3, with probability 1, it is possible to choose l sufficiently large such that for $n > l$, and $\forall a, x, y$,

$$|P_{xy}^{(n)}[a] - P_{xy}| < \frac{\epsilon}{3s(s + 1)\mathcal{R}}, \text{ and } |\mathcal{R}_x^{(n)}(a) - \mathcal{R}_x(a)| < \frac{\epsilon}{3s(s + 1)},$$

By B.2, choose h sufficiently large such that for $n > h$, the probability, after taking s actions, of ending up at a level lower than l is less than $\min\{(\epsilon(1 - \gamma)/6s\mathcal{R}), (\epsilon/3s(s + 1)\mathcal{R})\}$. This means that

$$|P_{xy}^{\prime(n)}[a] - P_{xy}| < \frac{2\epsilon}{3s(s+1)\mathcal{R}}, \text{ and } |\mathcal{R}_x^{\prime(n)}(a) - \mathcal{R}_x(a)| < \frac{2\epsilon}{3s(s+1)},$$

where the primes on $P^{\prime(n)}$ and $\mathcal{R}^{\prime(n)}$ indicate that these are conditional on the level in the ARP after the s^{th} step being greater than l.

Then, for $n > h$, by B.4, compare the value $\bar{\mathcal{Q}}_{\text{ARP}}(\langle x, n\rangle, a_1, \ldots, a_s)$ of taking actions $a_1, \ldots, a_s$ at state x in the ARP, with $\bar{\mathcal{Q}}(x, a_1, \ldots, a_s)$ of taking them in the real process:[3]

$$|\bar{\mathcal{Q}}_{\text{ARP}}(\langle x, n\rangle, a_1, \ldots, a_s) - \bar{\mathcal{Q}}(x, a_1, \ldots, a_s)| < \frac{\epsilon(1-\gamma)}{6s\mathcal{R}} \frac{2s\mathcal{R}}{1-\gamma} + \frac{2\epsilon}{3s(s+1)} \frac{s(s+1)}{2} = \frac{2\epsilon}{3}. \quad (4)$$

Where, in equation 4, the first term counts the cost of conditions for B.2 not holding, as the cost of straying below l is bounded by $2s\mathcal{R}/(1-\gamma)$. The second term is the cost, from B.4, of the incorrect rewards and transition probabilities.

However, by B.1, the effect of taking only s actions makes a difference of less than $\epsilon/6$ for both the ARP and the real process. Also since equation 4 applies to any set of actions, it applies perforce to a set of actions optimal for *either* the ARP or the real process. Therefore

$$|\mathcal{Q}^*_{\text{ARP}}(\langle x, n\rangle, a) - \mathcal{Q}^*(x, a)| < \epsilon.$$

So, with probability 1, $\mathcal{Q}_n(x, a) \to \mathcal{Q}^*(x, a)$ as $n \to \infty$ as required.

4. Discussions and conclusions

For the sake of clarity, the theorem proved above was somewhat restricted. Two particular extensions to the version of Q-learning described above have been used in practice. One is the non-discounted case ($\gamma = 1$), but for a Markov process with absorbing goal states, and the other is to the case where many of the $\mathcal{Q}$ values are updated in each iteration rather than just one (Barto, Bradtke & Singh, 1991). The convergence result holds for both of these, and this section sketches the modifications to the proof that are necessary.

A process with absorbing goal states has one or more states which are bound in the end to trap the agent. This ultimate certainty of being trapped plays the rôle that $\gamma < 1$ played in the earlier proof, in ensuring that the value of state x under any policy π, $V^\pi(x)$, is bounded, and that lemma B.1 holds, i.e., that the difference between considering infinite and finite (s) numbers of actions tends to 0 as $s \to \infty$.

Since the process would always get trapped were it allowed to run, for every state x there is some number of actions $u(x)$ such that no matter what they are, there is a probability $p(x) > 0$ of having reached one of the goal states after executing those actions. Take

$u^* = \max_x \{u(x)\}$, and $p^* = \min_x \{p(x)\} > 0$ (since there is only a finite number of states). Then a crude upper bound for $V^\pi(x)$ is

$$|V^\pi(x)| \leq u^*\mathcal{R} + (1 - p^*)u^*\mathcal{R} + (1 - p^*)^2u^*\mathcal{R} + \ldots$$

$$= \frac{u^*\mathcal{R}}{p^*}$$

since in each u^* steps the agent earns a reward of less than $u^*\mathcal{R}$, and has probability less than $(1 - p^*)$ of not having been trapped. Similarly, the effect of measuring the reward after only ϕu^* steps is less than $(1 - p^*)^\phi u^*\mathcal{R} \to 0$ as $\phi \to \infty$, and so an equivalent of lemma B.1 does hold.

Changing more than one $\mathcal{Q}$ value on each iteration requires a minor modification to the action replay process ARP such that an action can be taken at any level at which it was executed in the real process—i.e., more than one action can be taken at each level. As long as the stochastic convergence conditions in equation 3 are still satisfied, the proof requires no non-trivial modification. The $\mathcal{Q}_n(x, a)$ values are still optimal for the modified ARP, and this still tends to the real process in the original manner. Intuitively, the proof relies on the ARP estimating rewards and transition functions based on many episodes, and this is just speeded up by changing more than one $\mathcal{Q}$ value per iteration.

Although the paper has so far presented an apparent dichotomy between $\mathcal{Q}$-learning and methods based on certainty equivalence, such as Sato, Abe and Takeda (1988), in fact there is more of a continuum. If the agent can remember the details of its learning episodes, then, after altering the learning rates, it can use each of them more than once (which is equivalent to putting cards that were thrown away, back in, lower down on the ARP stack). This biases the $\mathcal{Q}$-learning process towards the particular sample of the rewards and transitions that it has experienced. In the limit of re-presenting 'old' cards infinitely often, this reuse amounts to the certainty equivalence step of calculating the optimal actions for the observed sample of the Markovian environment rather than the actual environment itself.

The theorem above only proves the convergence of a restricted version of Watkins' (1989) comprehensive $\mathcal{Q}$-learning algorithm, since it does not permit updates based on the rewards from more than one iteration. This addition was pioneered by Sutton (1984; 1988) in his TD(λ) algorithm, in which a reward from a step taken r iterations previously is weighted by λ^r, where $\lambda < 1$. Unfortunately, the theorem does not extend trivially to this case, and alternative proof methods such as those in Kushner and Clark (1978) may be required.

This paper has presented the proof outlined by Watkins (1989) that $\mathcal{Q}$-learning converges with probability one under reasonable conditions on the learning rates and the Markovian environment. Such a guarantee has previously eluded most methods of reinforcement learning.

Acknowledgments

We are very grateful to Andy Barto, Graeme Mitchison, Steve Nowlan, Satinder Singh, Rich Sutton and three anonymous reviewers for their valuable comments on multifarious aspects of $\mathcal{Q}$-learning and this paper. Such clarity as it possesses owes to Rich Sutton's

tireless efforts. Support was from Philips Research Laboratories and SERC. PD's current address is CNL, The Salk Institute, PO Box 85800, San Diego, CA 92186-5800, USA.

Notes

1. In general, the set of available actions may differ from state to state. Here we assume it does not, to simplify the notation. The theorem we present can straightfowardly be extended to the general case.
2. The discount factor for the ARP will be taken to be γ, the same as for the real process.
3. The bars over the $\mathcal{Q}$ indicate that the sum is over only a finite number of actions, with 0 terminal reward.

Appendix

The action-replay process

The definition of the ARP is contingent on a particular sequence of episodes observed in the real process. The state space of the ARP is $\{\langle x, n\rangle\}$, for x a state of the real process and $n \geq 1$, together with one, special, absorbing state, and the action space is $\{a\}$ for a an action from the real process.

The stochastic reward and state transition consequent on performing action a at state $\langle x, n\rangle$ is given as follows. For convenience, define $n^i \equiv n^i(x, a)$, as the index of the i^{th} time action a was tried at state x. Define

$$i_* = \begin{cases} \text{argmax}_i \{n^i < n\} & \text{if } x, a \text{ has been executed before episode } n \\ 0 & \text{otherwise} \end{cases}$$

such that n^{i_*} is the last time before episode n that x, a was exeucted in the real process. If $i_* = 0$, the reward is set as $\mathcal{Q}_0(x, a)$, and the ARP absorbs. Otherwise, let

$$i_e = \begin{cases} i_* & \text{with probablity } \alpha_{n^{i_*}} \\ i_* - 1 & \text{with probability } (1 - \alpha_{n^{i_*}})\alpha_{n^{i_*-1}} \\ i_* - 2 & \text{with probability } (1 - \alpha_{n^{i_*}})(1 - \alpha_{n^{i_*-1}})\alpha_{n^{i_*-2}}, \\ & \vdots \\ 0 & \text{with probability } \prod_{i=1}^{i_*} (1 - \alpha_{n^i}) \end{cases}$$

be the index of the episode that is *replayed* or taken, chosen probabilistically from the collection of existing samples from the real process. If $i_e = 0$, then the reward is set at $\mathcal{Q}_0(x, a)$ and the ARP absorbs, as above, Otherwise, taking i_e provides reward $r_{n^{i_e}}$, and causes a state transition to $\langle y_{n^{i_e}}, n^{i_e} - 1\rangle$ which is at level $n^{i_e} - 1$. This last point is crucial, taking an action in the ARP always causes a state transition to a lower level—so it ultimately terminates. The discount factor in the ARP is γ, the same as in the real process.

Lemma A: $\mathcal{Q}_n$ are optimal for the ARP

$\mathcal{Q}_n(x, a)$ are the optimal action values for ARP states $\langle x, n\rangle$ and ARP actions a. That is

$$\mathcal{Q}_n(x, a) = \mathcal{Q}^*_{\mathsf{ARP}}(\langle x, n\rangle, a), \ \forall a, x, \text{ and } n \geq 0.$$

Proof

By induction. From the construction of the ARP, $\mathcal{Q}_0(x, a)$ is the optimal—indeed the only possible—action value of $\langle x, 0\rangle$, a. Therefore,

$$\mathcal{Q}_0(x, a) = \mathcal{Q}^*_{\mathsf{ARP}}(\langle x, 0\rangle, a).$$

Hence the theorem holds for $n = 0$.

Suppose that the values of $\mathcal{Q}_{n-1}$, as produced by the $\mathcal{Q}$-learning rule, are the optimal action values for the ARP at level $n - 1$, that is

$$\mathcal{Q}_{n-1}(x, a) = \mathcal{Q}^*_{\mathsf{ARP}}(\langle x, n - 1\rangle, a), \ \forall a, x.$$

This implies that the $V_{n-1}(x)$ are the optimal values V^* for the ARP at the $n - 1^{\text{th}}$ level, that is

$$V^*(\langle x, n - 1\rangle) = V_{n-1}(x) \equiv \max_a \mathcal{Q}_{n-1}(x, a).$$

Now consider the cases in trying to perform action a in $\langle x, n\rangle$. If $x, a \neq x_n, a_n$, then this is the same as performing a in $\langle x, n - 1\rangle$, and $\mathcal{Q}_n(x, a) = \mathcal{Q}_{n-1}(x, a)$. Therefore,

$$\mathcal{Q}_n(x, a) = \mathcal{Q}_{n-1}(x, a) = \mathcal{Q}^*_{\mathsf{ARP}}(\langle x, n - 1\rangle, a) = \mathcal{Q}^*_{\mathsf{ARP}}(\langle x, n\rangle, a)$$

Otherwise, performing a_n in $\langle x_n, n\rangle$

- with probability $1 - \alpha_n$ is exactly the same as performing a_n in $\langle x_n, n - 1\rangle$, or
- with probability α_n yields immediate reward r_n and new state $\langle y_n, n - 1\rangle$.

Therefore the optimal action value in the ARP of $\langle x_n, n\rangle$, a_n is

$$\begin{aligned}\mathcal{Q}^*_{\mathsf{ARP}}(\langle x_n, n\rangle, a_n) &= (1 - \alpha_n)\mathcal{Q}^*_{\mathsf{ARP}}(\langle x_n, n - 1\rangle, a_n) + \alpha_n(r_n + \gamma V^*(\langle y_n, n - 1\rangle)) \\ &= (1 - \alpha_n)\mathcal{Q}_{n-1}(x_n, a_n) + \alpha_n(r_n + \gamma V_{n-1}(y_n)) \\ &= \mathcal{Q}_n(x_n, a_n)\end{aligned}$$

from the induction hypothesis and the $\mathcal{Q}_n$ interation formula in equation 1.

Hence, $\mathcal{Q}_n(x, a) = \mathcal{Q}^*_{\mathsf{ARP}}(\langle x, n\rangle, a)$, $\forall a, x$, as required.

Lemma B

B.1 Discounting infinite sequences

Consider a discounted, bounded-reward, finite Markov process with transition matrix $P_{xy}[a]$. From any starting state x, the difference between the value of that state under any set of s actions and under those same s actions followed by any arbitrary policy tends to 0 as $s \to \infty$.

Proof

Ignoring the value of the $s + 1^{\text{th}}$ state incurs a penalty of

$$\delta \equiv \gamma^s \sum_{y_{s+1}} P_{y_s y_{s+1}}[a_s] V^\pi(y_{s+1})$$

But if all rewards are bounded by $\mathcal{R}$, $|V^\pi(x)| < \mathcal{R}/(1 - \gamma)$, and so

$$|\delta| < \gamma^s \frac{\mathcal{R}}{1 - \gamma} \to 0 \text{ as } s \to \infty.$$

B.2 The probability of straying below level l is executing s *actions can be make arbitrarily small*

Given any level l, there exists another yet higher level, h, such that the probability can be made arbitrarily small of straying below l after taking s actions in the ARP, starting from above h.

Proof

Define i_h as the largest i such that $n^i(x, a) \leq n$, and i_l as the smallest such that $n^i(x, a) \geq l$. Then, defining $\alpha_{n^0} = 1$, the probability of straying below l starting from $\langle x, n \rangle$, $n > l$ executing action a is:

$$\left[\prod_{i=i_l}^{i_h} (1 - \alpha_{n^i}) \right] \sum_{j=0}^{i_l - 1} \left\{ \alpha_{n^j} \prod_{k=j+1}^{i_l - 1} (1 - \alpha_{n^k}) \right\} \leq \prod_{i=i_l}^{i_h} (1 - \alpha_{n^i}),$$

where, as before, $n^i \equiv n^i(x, a)$. But $\Pi_{i=i_l}^{i_h}(1 - \alpha_{n^i}) < \exp(-\Sigma_{i=i_l}^{i_h} \alpha_{n^i}) \to 0$ as n and hence $i_h \to \infty$. Furthermore, since the state and action spaces are finite, given η, there exists some level n_1 such that starting above there from any (x, a) leads to a level above l with probability at least $1 - \eta$. This argument iterates for the second action with n_1 as the new lower limit. η can be chosen appropriately to set the overall probability of straying below l less than any arbitrary $\epsilon > 0$.

B.3 Rewards and transition probabilities converge with probabability 1

With probability 1, the probabilities $P_{xy}^{(n)}[a]$ and expected rewards $\mathfrak{R}_x^{(n)}(a)$ in the ARP converge and tend to the transition matrices and expected rewards in the real process as the level n increases to infinity.

Proof

A standard theorem in stochastic convergence (e.g., theorem 2.3.1 of Kushner & Clark, 1978) states that if X_n are updated according to

$$X_{n+1} = X_n + \beta_n(\xi_n - X_n)$$

where $0 \leq \beta_n < 1$, $\Sigma_{i=1}^{\infty} \beta_n = \infty$, $\Sigma_{i=1}^{\infty} \beta_n^2 < \infty$, and ξ_n are bounded random variables with mean Ξ, then

$$X_n \to \Xi, \text{ as } n \to \infty, \text{ with probability 1.}$$

If $\mathfrak{R}_{\langle x,n\rangle}(a)$ is the expected immediate reward for performing action a from state x at level n in the ARP, then $\mathfrak{R}_{\langle x,n\rangle}(a)$ satisfies

$$\mathfrak{R}_{\langle x,n^{i+1}\rangle}(a) = \mathfrak{R}_{\langle x,n^i\rangle}(a) + \alpha_{n^{i+1}}(r_{n^{i+1}} - \mathfrak{R}_{\langle x,n^i\rangle}(a))$$

where the $\mathfrak{R}$ and the α satisfy the conditions of the theorem with $\Xi = \mathfrak{R}_x(a)$, and remembering that n^i is the i^{th} occasion on which action a was tried at state x. Therefore $\mathfrak{R}_{\langle x,n\rangle}(a) \to \mathfrak{R}_x(a)$ as $n \to \infty$, with probbility one. Also, since there is only a finite number of states and actions, the convergence is uniform.

Similarly, define

$$\chi_n(y) = \begin{cases} 1 & \text{if } y_n = y \\ 0 & \text{otherwise} \end{cases}$$

as a (random variable) indicator function of the n^{th} transition, mean value $P_{xy}(a)$. Then, with $P_{xy}^{(n)}[a]$ as the probability of ending up at state y based on a transition from state x using action a at level n in the ARP,

$$P_{xy}^{(n^{i+1})}[a] = P_{xy}^{(n^i)}[a] + \alpha_{n^{i+1}}(\chi_{n^{i+1}} - P_{xy}^{(n^i)}[a]),$$

and so, by the theorem, $P_{xy}^{(n)}[a] \to P_{xy}[a]$ (the transition matrix in the real process) as $n \to \infty$, with probability one.

Since, in addition, all observations from the real process are independent, and, by B.2, the probability of straying below a fixed level k can be made arbitrarily small, the transition probabilities and expected rewards for a single step *conditional* on ending up at a level greater than k also converge to $P_{xy}[a]$ and $\mathfrak{R}_x(a)$ as $n \to \infty$.

B.4 Close rewards and transitions imply close values

Let $P^i_{xy}[a]$, for $i = 1 \ldots s$ be the transition matrices of s Markov chains, and $\mathcal{R}^i_x(a)$ be the reward functions. Consider the s-step chain formed from the concatenation of these—i.e., starting from state x_1, move to state x_2 according to $P^1_{x_1x_2}[a_1]$, then state x_3 according to $P^2_{x_2x_3}[a_2]$, and so on, with commensurate rewards. Given $\eta > 0$, if $P^i[a]$ are within $\eta/\mathcal{R}$ of $P_{xy}[a]$, $\forall a, x, y$, and $\mathcal{R}^1_x(a) \ldots \mathcal{R}^s_x(a)$ are within η of $\mathcal{R}_x(a)$, $\forall a, x$, then the value of the s actions in the concatenated chain is within $\eta s(s + 1)/2$ of their value in the real process.

Proof

Define:

$$\bar{\mathcal{Q}}(x, a_1, a_2) = \mathcal{R}_x(a_1) + \gamma \sum_y P_{xy}[a_1]\mathcal{R}_y(a_2)$$

as the expected reward in the real process for executing two actions, a_1 and a_2 at state x, and

$$\bar{\mathcal{Q}}'(x, a_1, a_2) = \mathcal{R}^1_x(a_1) + \gamma \sum_y P^1_{xy}[a_1]\mathcal{R}^2_y(a_2)$$

as the equivalent in the concatenated chain for exactly the same actions.

Then, since $|\mathcal{R}^i_x(a) - \mathcal{R}_x(a)| < \eta$ and $P^i_{xy}[a] - P_{xy}[a]| < \eta/\mathcal{R}$, $\forall a, i, x, y$,

$$\begin{aligned}|\bar{\mathcal{Q}}'(x, a_1, a_2) - \bar{\mathcal{Q}}(x, a_1, a_2)| \leq\ & |\mathcal{R}^1_x(a_1) - \mathcal{R}_x(a_1)| + \\ & \gamma \left| \sum_y P^2_{xy}[a_2](\mathcal{R}^2_y(a_2) - \mathcal{R}_x(a_2)) \right| + \\ & \gamma \left| \sum_y (P^2_{xy}[a_2] - P_{xy}[a_2])\mathcal{R}_x(a_2) \right| \\ <\ & 3\eta\end{aligned}$$

Similarly, for s actions,

$$|\bar{\mathcal{Q}}'(x, a_1, \ldots, a_s) - \bar{\mathcal{Q}}(x, a_1, \ldots, a_s)| < \frac{s(s + 1)}{2}\eta.$$

This applies to the ARP if the rewards and transition matrices at the successively lower levels are sufficiently close to those in the real process—the main body of the theorem quantifies the cost of this condition failing.

References

Barto, A.G., Bradtke, S.J. & Singh, S.P. (1991). *Real-time learning and control using asynchronous dynamic programming.* (COINS technical report 91-57). Amherst: University of Massachusetts.

Barto, A.G. & Singh, S.P. (1990). On the computational economics of reinforcement learning. In D.S. Touretzky, J. Elman, T.J. Sejnowski & G.E. Hinton, (Eds.), *Proceedings of the 1990 Connectionist Models Summer School.* San Mateo, CA: Morgan Kaufmann.

Bellman, R.E. & Dreyfus, S.E. (1962). *Applied dynamic programming.* RAND Corporation.

Chapman, D. & Kaelbling, L.P. (1991). Input generalization in delayed reinforcement learning: An algorithm and performance comparisons. *Proceedings of the 1991 International Joint Conference on Artificial Intelligence* (pp. 726-731).

Kushner, H. & Clark, D. (1978). *Stochastic approximation methods for constrained and unconstrained systems.* Berlin, Germany: Springer-Verlag.

Lin, L. (1992). Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning, 8.*

Mahadevan & Connell (1991). Automatic programming of behavior-based robots using reinforcement learning. *Proceedings of the 1991 National Conference on AI* (pp. 768-773).

Ross, S. (1983). *Introduction to stochastic dynamic programming.* New York, Academic Press.

Sato, M., Abe, K. & Takeda, H. (1988). Learning control of finite Markov chains with explicit trade-off between estimation and control. *IEEE Transactions on Systems, Man and Cybernetics, 18,* pp. 677-684.

Sutton, R.S. (1984). *Temporal credit assignment in reinforcement learning.* PhD Thesis, University of Massachusetts, Amherst, MA.

Sutton, R.S. (1988). Learning to predict by the methods of temporal difference. *Machine Learning, 3,* pp. 9-44.

Sutton. R.S. (1990). Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. *Proceedings of the Seventh International Conference on Machine Learning.* San Mateo, CA: Morgan Kaufmann.

Watkins, C.J.C.H. (1989). *Learning from delayed rewards.* PhD Thesis, University of Cambridge, England.

Werbos, P.J. (1977). Advanced forecasting methods for global crisis warning and models of intelligence. *General Systems Yearbook, 22,* pp. 25-38.