

Comparing Value-Function Estimation Algorithms in Undiscounted Problems

Ferenc Beleznyai
Department of Analysis
Eötvös Lóránd University
Budapest HUNGARY - 1088
Múzeum krt. 6–8.
beleznyai@cs.elte.hu

Tamás Gröbner
Mindmaker Ltd.
Budapest HUNGARY - 1121
Konkoly-Thege M. u. 29-33.
grobner@mindmaker.hu

Csaba Szepesvári*
Mindmaker Ltd.
Budapest HUNGARY - 1121
Konkoly-Thege M. u. 29-33.
szepes@mindmaker.hu

Abstract

We compare scaling properties of several value-function estimation algorithms. In particular, we prove that Q-learning can scale exponentially slowly with the number of states. We identify the reasons of the slow convergence and show that both TD(λ) and learning with a fixed learning-rate enjoy rather fast convergence, just like the model-based method.

1 Introduction

Recently, there was some discussion about if the indirect (model-based) or the direct (model-free) methods of reinforcement learning are more advantageous. Of course, the answer is affected by many factors. One such factor is the learning-speed measured in terms of the number of samples required to learn a good approximation of the optimal Q-function (with a given tolerance and with high probability). In this article we take this number as the basis of our measure of convergence rate.

In their recent paper Kearns and Singh argued that Q-learning is not much slower than model-based learning [2], namely that their sample complexities scale the same in terms of the number of states. The purpose of this short note is to show that this claim depends crucially on the presence of discounting. In particular, we give a simple example of an *undiscounted* Markov decision process when *on-line* Q-learning arguably converges much slower than other methods, including model-based learning.

The example is a special Markov reward process (there are no actions) in which the states are linearly ordered. The last, absorbing state yields a unit reward and there are no rewards otherwise. It turns out that one can prove that for the case of this example Q-learning has an exponentially slow convergence in the number of states: it takes at least $C_\varepsilon \exp(N)$ steps until the Q-value of the first state comes within an ε neighborhood of its limit value. This is illustrated in Figure 1. The slow convergence of Q-values in this example can be attributed to a number of reasons; (i) the "wrong" ordering of updates, (ii) that the learning rates converge to zero and (iii) that there is no discounting. *The example generalizes easily to more complex problems and shows that on-line*

* Homepage <http://victoria.mindmaker.hu/~szepes>

Q-learning can be very slow in problems having long chains in it. On the other hand, it is trivial to see that the model-based approach would converge in time linear in the size of the problem for this type of problems.

There are a number of other ways to avoid the slow convergence. By using a trajectory-based learning with reversed order updates the slow convergence phenomena disappears. In order to implement this strategy one needs to remember individual trajectories which may be prohibiting when the expected time of reaching an absorbing state is large. The resulting algorithm was criticized e.g. by Rummery and Niranjan [6] who argue that on-line Q-learning's advantage can be huge since it uses experience earlier and therefore (by their argument) more efficiently. Our example shows that this conclusion should be taken warily. It is certainly true that off-line (or trajectory-based) learning cannot be applied unless there is some kind of notion of trial boundaries. Therefore it is important to identify other ways to overcome the slow convergence of the on-line method.

One such way is to use a *fixed learning-rate*, a strategy often applied in practice. In general, using a fixed learning-rate may prevent the convergence of the estimates to their limiting values if the reward process is noisy. However, as it is proven in the article, for noiseless reward processes it can radically increase the rate of convergence to superquadratic. If convergence for noisy processes is required then one can compute the averages of the estimated values in a second step – this method is called Poljak-averaging and has been considered by a number of authors [5, 4].

Since here we are dealing with estimating the value of a fixed policy (there are no choices in our example) it is also reasonable to compare the performance of Q-learning to that of TD(λ). TD(λ) with $\lambda > 0$ can be expected to perform better than Q-learning since for $\lambda = 1$ it is equivalent with reverse-order updates.¹ Indeed, we found in the computer experiments that it performed better than Q-learning. In particular, if λ is scaled with the number of states than the exponentially slow convergence can be improved to become linear.

Another source of the slow convergence is that we used no discounting in our example. Indeed, this makes a big difference since if discounting is used then for a fixed ε ($0 < \varepsilon < 1$) the effective "chain length" is

$$N_{\text{eff}} = \log((1 - \gamma)\varepsilon / R_{\text{max}}) / \log(\gamma),$$

where R_{max} is an upper bound on the absolute value of the immediate rewards. The effective chain length means that trajectories not longer than N_{eff} are sufficient to determine the total expected discounted value of any policy. This can be proved easily in the following way: For any controlled reward-stream $\{r_t\}$ with initial state x_0 ,

$$\left| E \left[\sum_{t=0}^{\infty} \gamma^t r_t | x_0 \right] - E \left[\sum_{t=0}^{N_{\text{eff}}-1} \gamma^t r_t | x_0 \right] \right| \leq \gamma^{N_{\text{eff}}} R_{\text{max}} / (1 - \gamma).$$

Note that N_{eff} does not depend on the number of states. Indeed, this property of discounted problems was exploited in the paper of Kearns and Singh [2] and also in the paper of Kearns et. al [1].

The effective chain-length can be interpreted in different ways, depending on the interpretation of discounting. If the discounting is part of the problem definition *per se* (e.g. it is modeling a fixed inflation rate in economic problems) then it is meaningful for the decision-maker to fix a precision and compute everything up to the effective chain-length. If, however, the decision-maker is primarily interested in the limiting case $\gamma = 1$ and the discount-factor is used as a tool to ensure that the particular algorithms work (i.e. to ensure boundedness) then the algorithm designer must choose a discount-factor close enough to 1 so that the class of discounted optimal policies and the undiscounted optimal policies coincide. In general, this means that N_{eff} should scale with the size of the state space. This means that there is indeed a big conceptual difference between discounted and undiscounted Markovian decision problems.

¹We are considering trajectory-based TD(λ) only.

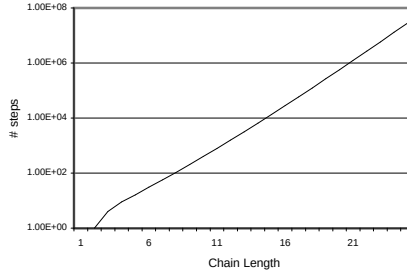


Figure 1: **Sample complexity of Q-learning** Note the logarithmic scale of the y axis.

2 Results

We assume that readers are familiar with reinforcement learning. First we describe the example we are using throughout the paper. Consider a Markov decision process with one action per state, consisting of a linearly ordered chain of N states: in each state there is only one action, the one leading to the next state. The process terminates upon reaching the N th state. All transitions result in zero reinforcement, except for the last one leading from state $N - 1$ to state N whose associated value equals to one.² Since there is only one action per state, the Q -values will be indexed only by their corresponding state, i.e., we write $Q(i)$ for $Q(i, a)$, where $a \in A = \{a\}$ is the single action available.

2.1 Analysis of Q-learning

In the on-line version of Q-learning reinforcement propagates from the state $N - 1$ in the backwards direction. The question considered here is how long does it take for the value of the first state $Q(1)$ to reach a fixed limit, say $1 - \varepsilon$ as a function of N . We shall assume that all Q values start at zero. In this simple example it is easy to see that the Q-learning algorithm ensures that all Q -values will monotonously increase to their limit of 1. In our case the Q-learning equations reduce to

$$Q_0(s) = 0, \quad 1 \leq s \leq N \quad (1)$$

$$Q_{t+1}(s_t) = (1 - \alpha_t(s_t))Q_t(s_t) + \alpha_t(s_t)(r(s_t) + Q_t(s_{t+1})), \quad t \geq 0 \quad (2)$$

$$Q_t(N) = 0, \quad t \geq 0, \quad (3)$$

where $r(N - 1) = 1$, $r(s) = 0$ $s \neq N - 1$, $s_t \equiv t + 1 \pmod{N}$, and $\alpha_t(s)$ is the inverse of the number of times the state s has been visited up to time step t . It follows that during the k^{th} trial $\alpha_t(s)$ is $1/k$ for all states ($k = 1, 2, \dots$). Therefore, by relabelling the Q -values such that $Q_k(s)$ becomes the value of state s during the k^{th} trial we get

$$Q_k(s) = \left(1 - \frac{1}{k}\right) Q_{k-1}(s) + \frac{1}{k} Q_{k-1}(s + 1), \quad s < N - 1,$$

which is the familiar formula of averaging the values of the sequence $\{Q_k(s + 1)\}_k$ in an incremental way and storing the result in $Q_k(s)$:

$$Q_k(s) = \frac{1}{k} \sum_{i=0}^{k-1} Q_i(s + 1), \quad s < N - 1. \quad (4)$$

The boundary conditions are given by the equations $Q_k(N) = 0$ if $k \geq 1$ and $Q_k(N - 1) = 1$ if $k \geq 1$.³

²This is assumed just to simplify the presentation. The analysis can be repeated without any effort for other similar reward structures.

³Therefore the claim of slow convergence is transformed to the claim that the Cesaro averages of the Cesaro averages of .. (etc.) the Cesaro averages of the sequence $\{0, 1, 1, 1, 1, \dots\}$ admit a slow convergence.

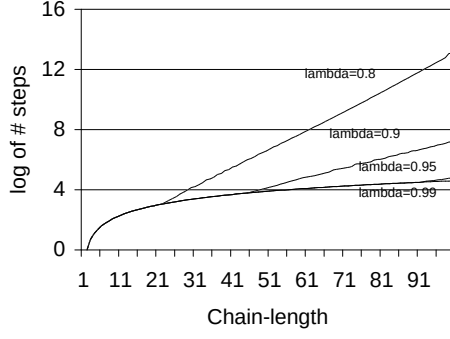


Figure 2: **Sample complexity of TD(λ) for different λ values.** Note the logarithmic scale of the y axis.

Now, we would like to develop an upper bound for $Q_k(1)$. Firstly, note that $\{Q_k(s)\}_k$ and $\{Q_k(\cdot)\}$ are monotone increasing and $Q_k(s) \leq 1$ for all k and s . Let

$$S_k = \frac{Q_k(1) + \dots + Q_k(N-2)}{N-2}.$$

Then $Q_k(1) < S_k$, $S_0 = 0$ and

$$S_k = S_{k-1} + \frac{Q_{k-1}(N-1) - Q_{k-1}(1)}{k(N-2)} \leq S_{k-1} + \frac{Q_{k-1}(N-1)}{k(N-2)} = S_{k-1} + \frac{1}{k(N-2)},$$

and thus

$$Q_k(1) < S_k \leq \frac{1}{N-2} \sum_{j=1}^k \frac{1}{j} < \frac{1 + \log k}{N-2}.$$

Therefore in order to have $1 - \varepsilon < Q_k(1)$ one needs

$$1 - \varepsilon < \frac{1 + \log k}{N-2},$$

i.e.

$$k > e^{(1-\varepsilon)(N-2)-1},$$

meaning that Q-learning is exponentially slow in this case.

The algorithm described above was simulated for different chain lengths N . The number of steps required for $Q_t(1)$ to reach 0.99 is plotted on a logarithmic plot in Figure 1. After a short initial transient, the plot is fairly linear, indicating that the convergence is exponentially slow in the size of the problem.

2.2 TD(λ)

The Q-learning algorithm discussed above corresponds to on-line TD(0). We also ran some computer simulations to investigate the scaling properties of TD(λ) for $\lambda > 0$. Again, we plotted the number of steps on a logarithmic plot for different N 's (see Figure 2). It turned out that, after an initial transient period, the convergence becomes exponentially slow for all $0 \leq \lambda < 1$. During the transient, however, the number of steps scales linearly with N , and this transient becomes infinitely long as $\lambda \rightarrow 1$ like $1/(1-\lambda)$. This is in good agreement with the trivial observation that the TD(1) algorithm is linear as it corresponds to straight Monte-Carlo estimation. Note that linear scaling can be achieved if one chooses $\lambda_N = 1 - 1/N$.

2.3 Constant Learning Rate

If we fix the learning rate α (i.e. $\alpha_t(s) = \alpha$ for all $t \geq 0$) then the algorithm becomes faster. Namely, we prove below that it becomes superquadratic ($t_N = \Omega(1/\alpha N^2)$).⁴ For $\alpha = 1$, it is obvious to see that the number of steps required to propagate the value 1 to the beginning of the chain is $(N-1)^2$. Now let $0 < \alpha < 1$. Then one can prove by induction that

$$Q_k(N-j+1) = \begin{cases} 0, & \text{if } k < j; \\ \alpha^j p_{N-j+1,k}, & \text{otherwise,} \end{cases}$$

where

$$p_{N-j+1,k} = \sum_{i=0}^{k-j} \binom{j+i-1}{i} (1-\alpha)^i.$$

Therefore

$$Q_k(1) = \alpha^N \sum_{i=0}^{k-N} \binom{N+i-1}{i} (1-\alpha)^i, \quad k \geq N.$$

Now, observe that the sum on the r.h.s. of the above equation is just the $(k-N)$ th Taylor-polynomial of the function $f(x) = (1-x)^{-N}$ around the point zero, evaluated at $x = 1-\alpha$ (indeed, $f^{(k)}(x) = (1-x)^{-(N+k)} \prod_{i=0}^{k-1} (N+i)$.) The rate of convergence of the Taylor-series

$$T_n(x) = \sum_{i=0}^n \binom{N+i-1}{i} x^i$$

to $f(x)$ can be estimated using the Cauchy-residual theorem. According to this theorem there exists a number $\xi \in [0, \bar{\alpha}]$ such that

$$|f(\bar{\alpha}) - T_n(\bar{\alpha})| \leq \left| \frac{f^{(n+1)}(\xi)}{(n+1)!} (\bar{\alpha} - \xi)^{n+1} \right|,$$

where $\bar{\alpha} = 1-\alpha$. Now, since the function $g(\xi) = (\bar{\alpha} - \xi)/(1-\xi)$ is decreasing on the interval $[0, \bar{\alpha}]$ and its maximum is taken on at $\xi = 0$ with $g(0) = \bar{\alpha}$. Therefore

$$|f(\bar{\alpha}) - T_n(\bar{\alpha})| \leq (1-\xi)^{-(N+1)} \bar{\alpha}^{n+1} \prod_{i=0}^n \frac{N+i}{i+1}.$$

The inequality

$$A_n = \prod_{i=0}^n \frac{N+i}{i+1} \leq (n+2)^{N-1}$$

can be easily proved if one takes the logarithm of A_n and uses the inequalities $\log(1+x) \leq x$ and $1 + 1/2 + \dots + 1/(n+1) \leq \int_1^{n+2} 1/x dx = \log(n+2)$. By means of the above inequalities we get that

$$\left| 1 - \alpha^N \sum_{i=0}^n \binom{N+i-1}{i} \bar{\alpha}^i \right| \leq A_n \bar{\alpha}^{n+1} \leq (n+2)^{N-1} \bar{\alpha}^{n+1}.$$

It follows that in order to achieve the ε error-bound, one needs at least $k_N = N + \hat{k}_N$ trials, where $\hat{k}_N / \log \hat{k}_N \geq N$. Since here k_N is the number of trials and one trial consists of N steps, one gets that the number of steps required to achieve the ε error-bound scales worse than N^2 and better than N^3 .

Note that the decay of the learning rate is required only to ensure that the effect of noise in the system is cancelled out asymptotically: in deterministic problems a small enough, fixed learning rate is sufficient to ensure convergence. In stochastic problems, another opportunity is to use Poljak averaging on the top of the fixed learning-rate scheme.

⁴In fact, from the analysis below it follows that $t_N = O(N^3)$.

3 Relation to Previous Work

In the context of *satisficing search*⁵ and deterministic problems Whitehead has shown that the number of steps that a zero-initialized Q-learning algorithm with goal-reward representation and greedy action selection needs in order to reach a goal state and terminate can be exponential in the number of states [8]. If, however, an action-penalty representation is used then the complexity of search drops to $O(|A|N)$ as shown by Koenig and Simmons [3] (they proved this only for deterministic problems). They also considered bi-directional versions of both Q-learning and the model-based algorithm for the construction of the optimal value (or action-value) function and show that they are no more complex than the problem of reaching a goal state from a particular initial state. Our conjecture that Q-learning can be exponentially slow also in deterministic problems does not contradict the results of Koenig and Simmons since the version of Q-learning they considered uses a fixed learning rate equal to 1, whilst the slow convergence conjecture depend crucially on the assumption that the learning-rates are decayed.

In an earlier paper one of the authors has proven that an upper bound on the asymptotic rate of convergence of Q-learning is $O(1/t^{R(1-\gamma)})$ provided that $R(1-\gamma) < 1/2$ and Q-learning is subject to the law of iterated logarithm otherwise [7]. This bound did not concern directly scaling with the number of states. Here R is a constant related to the “mixing-rate” of the aperiodic ergodic Markov chain corresponding to an appropriately fixed stationary policy. This result cannot be interpreted in MDPs where there are now policies yielding aperiodic and/or ergodic Markov chains. The MDP considered in this paper is such an environment. Note, however, that the the slow convergence shares the same root in both cases; the steady decay of the learning-rate towards zero which results in very small learning-rates before the relevant information “arrives” to a given state.

We conjecture that under the above conditions Q-learning indeed does not obeys the law of iterated logarithm, i.e. the above bound is essentially tight. Such a result would enable us to compare the asymptotic rates of convergence of Q-learning and model-based methods. To see this consider a model-based learning algorithm which runs value iteration (using the latest estimates of transition probabilities and rewards) until convergence after each observation. As it was noted in [7], such a model-based learning clearly satisfies the law of iterated logarithm. This is proven in two stpes: firstly, the estimation process of the transition probabilities and immediate rewards is subject to this law where the time scale is given by R (but scaling time linearly does not effect the form of the rate of convergence). Secondly, the L^∞ distance between the fixed point of the approximate Bellman operator and the optimal value function can be bounded from above by the distance of the approximated transition probabilities and the rewards to the true transition probabilities and rewards.

Therefore proving a polynomial lower-bound for Q-learning would enable one to conclude that Q-learning is slower than model-based learning if the underlying process is slowly mixing. This remains, however, a topic for further research.

4 Conclusions

In this paper we have shown that Q-learning applied to undiscounted problems with a decaying learning rate can exhibit exponentially slow convergence. We have identified a number of reasons of this negative result and considered various methods to overcome it. In particular, we have shown that slow convergence can be avoided in our example, if off-line (or trajectory based) learning is used or if TD(λ) is used with λ depending on the number of states or if a constant learning rate is used.

We think that the simple example presented in this paper brings some more insight into the various tradeoffs present in reinforcement learning methods. We believe that a similar lower bounds could also be carried out for the case of general optimization problems having “long chains” in it. We consider this as an important research topic. Another important problem is to give similar

⁵In satisficing search one is interested in finding a solution to a given problem. This is quite different from the problem considered in this paper, namely that of learning a value function.

bounds for the case of $TD(\lambda)$ and to analyze the convergence and convergence rate properties of Juditsky style algorithms.

5 Acknowledgements

This work was partially supported by OTKA grant no. F20132 and the Hungarian Ministry of Education grant No. FKFP1354/1997.

References

- [1] M. Kearns, Y. Mansour, and A.Y. Ng. A sparse sampling algorithm for near-optimal planning in large Markovian decision processes. In *Proceedings of IJCAI'99*, 1999.
- [2] M. Kearns and S. Singh. Finite-sample convergence rates for Q-learning and indirect algorithms. In *Advances in Neural Information Processing Systems 10*, 1998.
- [3] S. Koenig and R.G. Simmons. Complexity analysis of real-time reinforcement learning applied to finding shortest paths in deterministic domains. Technical Report CMU-CS-93-106, School of Computer Science, Carnegie Mellon University, 1993.
- [4] H.J. Kushner and G.G. Yin. *Stochastic Approximation Algorithms and Applications*. Springer-Verlag, New York, 1997.
- [5] B.T. Poljak and A.B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30:838–855, 1992.
- [6] G. A. Rummery and M. Niranjan. On-line Q-learning using connectionist systems. Technical Report CUED/F-INFENG/TR 166, Cambridge University Engineering Department, 1994.
- [7] Cs. Szepesvári. On the asymptotic convergence rate of Q-learning. In *Proc. of Neural Information Processing Systems*, 1997. in press.
- [8] S.D. Whitehead. Complexity and cooperation in Q-learning. In L.A. Birnbaum and G.C. Collins, editors, *Machine Learning: Proc. of the Eighth International Workshop*, pages 363–367. Morgan Kaufmann, San Mateo, CA, 1991.