6 Basic Convergence Results for RL Algorithms

We establish here some asymptotic convergence results for the basic RL algorithms, by showing that they reduce to Stochastic Approximation schemes. We focus on the discounted-cost problem, which is easiest. Analogous results exist for shortest-path problems.

We do not directly consider here the issue of *exploration*, which is essential for convergence to the optimal policy. Thus, where required we will simply *assume* that all actions are sampled often enough.

6.1 Q-learning

Recall the Q-learning algorithm, in the following generalized form:

$$Q_{n+1}(s,a) = Q_n(s,a) + \alpha_n(s,a)[r(s,a,s'_{s,a}) + \gamma \max_{a'} Q_n(s'_{s,a},a') - Q_n(s,a)]$$

where each $s'_{s,a}$ is determined randomly according to $p(s'|s,a)$.

We allow here $\alpha_n(s,a) = 0$, so that any number of (s,a) pairs can be updated at each stage.

This iteration can be viewed as an (asynchronous) Stochastic Approximation algorithm, with $Q \equiv \theta$. This leads to the following result.

Theorem 1 (Convergence of Q-learning).
Let $\gamma < 1$, and let Q^* be the optimal γ-discounted Q function. Assume

$$\sum_{h=0}^{\infty} \alpha_n(s,a) = \infty, \quad \sum_{n=0}^{\infty} \alpha_n(s,a)^2 < \infty \quad \text{(w.p. 1)} \quad \forall s, a\,.$$

Then

$$\lim_{n\to\infty} Q_n(s,a) = Q^*(s,a) \quad \text{(w.p. 1)} \quad \forall s, a\,.$$

Proof: Define the mapping H over the set of Q-functions as follows:

$$\begin{aligned}(HQ)(s,a) &= \sum_{s'} P(s'|s,a)[r(s,a,s') + \gamma \max_{a'} Q(s',a')] \\ &= E[r(s,a,s_{n+1}) + \gamma \max_{a'} Q(s_{n+1},a')|s_n = s, a_n = a]\,.\end{aligned}$$

The above Q-learning algorithm can thus be written in the standard SA form, with the noise vector ω_n given by:

$$\omega_n(s,a) = r(s,a,s'_{s,a}) + \gamma \max_{a'} Q_n(s'_{s,a},a') - (HQ)(s,a)\,.$$

We proceed to verify the assumptions in Prop. 4.4 of B.&T. (Theorem 2 in the previous chapter):

(a) Step-size requirements hold here by assumption.

(b) Noise Assumption N1: The definition of ω_n immediately implies that $E(\omega_n(s,a)|\mathcal{F}_n) = 0$. It is further easily seen that

$$E(\omega_n(s,a)^2|\mathcal{F}_n) \leq \text{ quadratic function of } \|Q\|_\infty\,.$$

(c) Contraction: As with the discounted DP operator, it may be verified that H is a γ-contraction w.r.t. the max-norm.

The required convergence result therefore follows by the above-mentioned theorem. $\square$

Remarks on basic (on-policy) Q-learning:

- In the basic version of the algorithm, we follow a state-action sequence $(s_n, a_n; n = 0, 1, \cdots)$ which is generated by some arbitrary policy, and at time n update $Q(s, a)$ only for $(s, a) = (s_n, a_n)$. This corresponds to the choice of gains:

$$\alpha_n(s, a) > 0 \quad \text{iff} \quad (s, a) = (s_n, a_n)\,.$$

- For $(s, a) = (s_n, a_n)$, a typical choice for α_n is

$$\alpha_n(s, a) = \hat{\alpha}(N_n(s, a))$$

  where N_n is the number of previous visits to (s, a), and $\hat{\alpha}(k)$ satisfies the standard assumptions.

- For the step-size requirements in the theorem to hold in this case it is required that each (s, a) pair is visited "relatively often". This should be verified by appropriate exploration policies!

Undiscounted case: Under appropriate "Stochastic Shortest Path" assumptions, it can be shown that H is a pseudo-contraction w.r.t. some weighted max-norm. Convergence follows as above.

6.2 Convergence of TD(λ)

TD(0) can be analyzed exactly as Q-learning learning. TD(λ) is slightly more involved. Recall the "on-line" version of TD(λ):

$$V_{n+1}(s) = V_n(s) + \alpha_n e_n(s) d_n \,, \quad s \in S$$

where

$$\begin{aligned} \alpha_n &= \text{gain} \\ e_n(s) &= \text{eligibility trace coefficient} \\ d_n &= r_n + \gamma V_n(s_{n+1}) - V_n(s_n) \\ \gamma &= \text{discount factor} \end{aligned}$$

Requirements on the Eligibility Trace

Several variants of the algorithm are obtained by different choices of $e_n(s)$, such as:

(a) First-visit TD(λ):

$$e_n(s) = (\gamma\lambda)^{n-m_1(s)} 1\{n \geq m_1(s)\} \,,$$

$m_1(s)$ is the time of first visit to state s (during the present run).

(b) Every-visit TD(λ):

$$e_n(s) = \sum_{j : m_j(s) \leq n} (\gamma\lambda)^{n-m_j(s)} \,,$$

$m_j(s)$ is the time of j^{th} visit to state s.

(c) First-visit with stopping:

$$e_n(s) = (\gamma\lambda)^{n-m_1(s)} 1\{m_1(s) \leq n \leq \tau\}$$

where τ is some stopping time – e.g., end of simulation run, or arrival to a state whose value $V(s)$ is known with high precision. $e_n(s)$ is restarted after τ.

A *general set of requirements* on the eligibility coefficients $e_n(s)$, which includes the above cases, is given as follows:

(a) $e_0(s) = 0$, $e_n(s) \geq 0$.

(b) $e_n(s) \leq \gamma e_{n-1}(s)$ if $s_n \neq s$,
$1 \leq e_n(s) \leq 1 + \gamma e_{n-1}(s)$ if $s_n = s$.

(c) $e_n(s)$ is measurable on the past.

Convergence

We now argue as follows.

- It may be seen that TD(λ) is in the form of the Stochastic Approximation algorithm, with $\theta_n \equiv V_n$, and

$$h(\theta) \equiv h(V) = (h(V)(s),\, s \in S)\,,$$

$$\begin{aligned} h(V)(x) &= E^{\pi}[d_n | V_n = V,\, s_n = s] \\ &= \sum_a \pi(a|s)[r(s,a) + \gamma \sum_{s'} p(s'|s,a)V(y)] - V(s) \\ &:= (HV)(s) - V(s)\,. \end{aligned}$$

  Here π is the *fixed stationary* policy that is used.

- For $0 < \gamma < 1$ it is obvious that H is a contraction operator.

- For convergence we now need to verify that the effective gains $\alpha_n e_n(s)$ satisfy the "usual assumptions". This may be verified by requiring that each state is visited "relatively often".

For $\gamma = 1$ a similar argument may be made for SSP (Stochastic Shortest Path) problems.

6.3 Actor-Critic Algorithms

Convergence of actor-critic type algorithms is harder to analyze and has been established more recently. We describe here some results from Konda and Borkar (2000).

Recall that the idea is to use a "fast" estimation loop to obtain $\hat{V}(s)$, and a slower loop to update the policy $\hat{\pi}$ given $\hat{V}$.

Let $V_n(s)$ and $\pi_n = (\pi_n(a|s))$ be the estimated value and policy at step n.

<u>Algorithm 1</u>

a. Value-function estimation (generalized TD(0)):

$$V_{n+1} = V_n(s) + \beta_n(s)[r(s, a_n(s)) + \gamma V_n(s_{n+1}(s)) - V_n(s)]\,, \quad s \in Y_n$$

where

Y_n – set of states updated at step n

$\beta_n(s)$ – gains

$s_{n+1}(s)$ – next state, chosen with distribution $p(s') = \sum_a p(s'|s,a)\pi_n(a,s)$.

b. Policy update:

$$\pi_{n+1}(a|s) = \pi_n(a|s) + \alpha_n(s,a)((\hat{Q}_n(s,a) - \hat{Q}_n(s,a_0)))\,, \quad (s,a) \in Z_n$$

where

Z_n – set of state-action pairs updated at step n

$\alpha_n(s,a)$ – gains

$\hat{Q}_n(s,a) := r(s,a) + \gamma V_n(s_{n+1}(s,a))$

$s_{n+1}(s,a)$ – next state, chosen according to $p(s'|s,a)$

a_0 – a fixed reference action (for each state).

b'. Policy normalization:

For each s, project the vector $(\pi_{n+1}(a|s),\ a \neq a_0)$ unto the following set of sub-probability vectors:

$$\{\pi : \pi(a) \geq 0,\ \sum_{a \neq a_0} \pi(a) \leq 1\}$$

and then let $\pi_{n+1}(a_0|s) = 1 - \sum_{a \neq a_0} \pi_{n+1}(a|s)$.

c. Rate requirements:

We first require that all updates are executed relatively often, namely that for some $\Delta > 0$,

$$\liminf_{n\to\infty} \frac{n_1(s)}{n} \geq \Delta\,, \quad \liminf_{n\to\infty} \frac{n_2(s,a)}{n} \geq \Delta\,,$$

where

$$n_1(s) = \sum_{k=1}^{n} 1\{s \in Y_k\}$$
$$n_2(s,a) = \sum_{k=1}^{n} 1\{(s,a) \in Z_k\}\,.$$

The gains are determined by some sequences $\alpha(m)$ and $\beta(m)$, as

$$\alpha_n(s,a) = \alpha(n_2(s,a))\,, \quad \beta_n(s) = \beta(n_1(s))\,.$$

The sequences $\alpha(m)$, $\beta(m)$ should satisfy:

(1) The standard summabilty assumptions.

(2) Policy updates are "slower": $\lim_{m\to\infty} \frac{\alpha(m)}{\beta(m)} = 0$.

(3) Some additional technical assumptions ...

All these requirements are satisfied, e.g., by $\alpha(m) = \frac{1}{m \log m}$, $\beta(m) = \frac{1}{m}$.

Under these assumptions, Algorithm 1 converges to the optimal value and policy.

Algorithm 2:

Same as Algorithm 1, except for the policy update (b):

$$\pi_{n+1}(a|s) = \pi_n(a|s) + a_n(s,a)[\{\hat{Q}_n(s,a) - V_n(s)\}\pi_n(a|s) + \xi_n(s,a)] \,.$$

$\xi_n(s,a)$ are sequences of "small" noise terms, these are needed to prevent the algorithm from getting stuck in the wrong "corners".

Algorithm 3:

Same as Algorithm 1, except for (b):

$$w_{n+1}(a|s) = w_n(a|s) + \alpha_n(s,a)[\hat{Q}_n(s,a) - V_n(s)]$$

and

$$\pi_n(s,a) := \frac{\exp(w_n(s,a))}{\sum_{a'} \exp(w_n(s,a'))} \,.$$

In all these variants, convergence is proved using a "two-time scale" Stochastic Approximation framework, the analysis is based on the ODE method which couples a "fast" ODE (for V) and a "slow" ODE (for π).