LINMA2725 Stochastic Optimal Control and Reinforcement Learning Part III

Course 2: Temporal Difference Techniques

Guillaume Berger

November 18, 2024

Reference: [1], Chapter 9.

Any questions or feedback are welcome.

Stochastic Optimal Control and Reinforcement Learning

Part III: Stochastic Systems

Guillaume Berger

Table of content

- Stochastic systems and stochastic control (1 course)
- Learning techniques for stochastic control (1-2 courses)
  - TD(λ)-learning: definition and analysis
  - Q-learning: definition and analysis
  - Next course: actor-critic methods
- Online learning techniques for stochastic control (1-2 courses)

Value and Q-functions approximation

Linear template:

$$h^{\theta}(x) = \theta^{\top}\psi(x)$$

where $\psi(x) = \big(\psi_1(x), \dots, \psi_d(x)\big)$ and $\theta \in \mathbb{R}^d$

Goal: find θ^* such that $h^{\theta^*} \approx h$

Similarly for the Q-function with $Q^{\theta}(x, u) = \theta^{\top}\psi(x, u)$

Temporal difference and Bellman error

In this course, we focus on the discounted cost, with $\gamma \in [0,1)$

Bellman error:

$$B_{n+1}^{\theta}(X) \stackrel{\text{def}}{=} -h^{\theta}\big(X(n)\big) + c\big(X(n)\big) + \gamma \mathbb{E}\big[\, h^{\theta}\big(X(n+1)\big) \mid X(n) \,\big]$$

Temporal difference:

$$D_{n+1}^{\theta}(X) \stackrel{\text{def}}{=} -h^{\theta}\big(X(n)\big) + c\big(X(n)\big) + \gamma h^{\theta}\big(X(n+1)\big)$$

Note: $B_n^{\theta}(X) = \mathbb{E}\big[\, D_{n+1}^{\theta}(X) \mid X(n) \,\big]$

Metrics for value function approximation

Mean-square Bellman error:

$$\min_{\theta} \mathbb{E}_\pi \left[\left(B^{\theta}_{n+1}(X) \right)^2 \right]$$

where the expectation is for a process X in steady state

Zero projected Bellman error (aka. Galerkin relaxation):

$$\mathbb{E}_\pi \left[D^{\theta}_{n+1}(X) \cdot \zeta_i(n) \right] = 0 \quad \forall i$$

where each ζ_i is a process in steady state

Boris G. Galerkin
(1871–1945)

The "π" in the subscript of the expectation "E" means that the processes are in steady state. For these processes, the definitions become independent of n. The subscript "π" will often be omitted in the notation in the following of this course.

Metrics for value function approximation

(continued)

Distance with true value function:

$$\min_{\theta} \left\| h^{\theta} - h \right\|_{\pi}$$

where h is the true value function and $\|f\|_{\pi}^{2} = \mathbb{E}_{\pi}\left[f\big(X(n)\big)^{2}\right]$

Mean-square Bellman error

Gradient descent: $\theta_{n+1} = \theta_n + \alpha_{n+1}\left(-\frac{1}{2}\nabla_\theta \mathbb{E}\left[\left(B^{\theta_n}(X)\right)^2\right]\right)$

Lemma 9.5. *The following holds for each* $\theta \in \mathbb{R}^d$*:*

$$-\tfrac{1}{2}\nabla_\theta \mathsf{E}_\pi[\{\mathcal{B}^\theta(X)\}^2] = \mathsf{E}_\pi[\mathcal{D}^\theta_{n+1}\zeta^\theta_n]$$

where

$$\zeta^\theta_n = \nabla_\theta \mathsf{E}[h^\theta(X(n)) - \gamma h^\theta(X(n+1)) \mid \mathcal{F}_n] \tag{9.30}$$

Stochastic gradient descent:

$$\theta_{n+1} = \theta_n + \alpha_{n+1}[\mathcal{D}^\theta_{n+1}\zeta^\theta_n]\big|_{\theta=\theta_n}$$

Conditional expectation

How to estimate $\zeta_n^\theta = \nabla_\theta \mathsf{E}[h^\theta(X(n)) - \gamma h^\theta(X(n+1)) \mid \mathcal{F}_n]$?

If X is finite: take $P(x'|x) \approx \frac{|\{k \le n | X(k+1)=x', X(k)=x\}|}{|\{k \le n | X(k)=x\}|}$

If X is infinite, the denominator is zero a.s.

By definition: $\mathbb{E}[\, Z \mid Y \,] \stackrel{\text{def}}{=} \arg \min_{Z'=g(Y)} \mathbb{E}[|Z' - Z|^2]$

Conditional expectation

(continued)

Approximated conditional expectation:

$$\widehat{\mathbb{E}}_{\hat{\psi}}\left[\, E^{\theta}_{n+1}(X) \mid X(n) \,\right] = \min_{E' = \hat{\theta}^{\top}\hat{\psi}(X(n))} \mathbb{E}\left[\left|E' - E^{\theta}_{n+1}(X)\right|^{2}\right]$$

It holds that $\hat{\theta}^{*} = A^{-1}b$ where

$$A = \mathbb{E}\left[\hat{\psi}\big(X(n)\big)\hat{\psi}\big(X(n)\big)^{\top}\right] \quad \text{and} \quad b = \mathbb{E}\left[\hat{\psi}\big(X(n)\big)E^{\theta}_{n+1}(X)\right]$$

Conditional expectation

(continued)

A and b can be approximated by

$$A \approx \hat{A}_k \stackrel{\text{def}}{=} \frac{1}{n} \sum_{k=0}^{n-1} \hat{\psi}\big(X(k)\big) \hat{\psi}\big(X(k)\big)^\top$$

$$b \approx \hat{b}_k \stackrel{\text{def}}{=} \frac{1}{n} \sum_{k=0}^{n-1} \hat{\psi}\big(X(k)\big) E_{k+1}^{\theta}(X)$$

Mean-square temporal difference

Gradient descent: $\theta_{n+1} = \theta_n + \alpha_{n+1}\left(-\frac{1}{2}\nabla_\theta \mathbb{E}\left[\left(D^{\theta_n}(X)\right)^2\right]\right)$

Stochastic gradient descent:

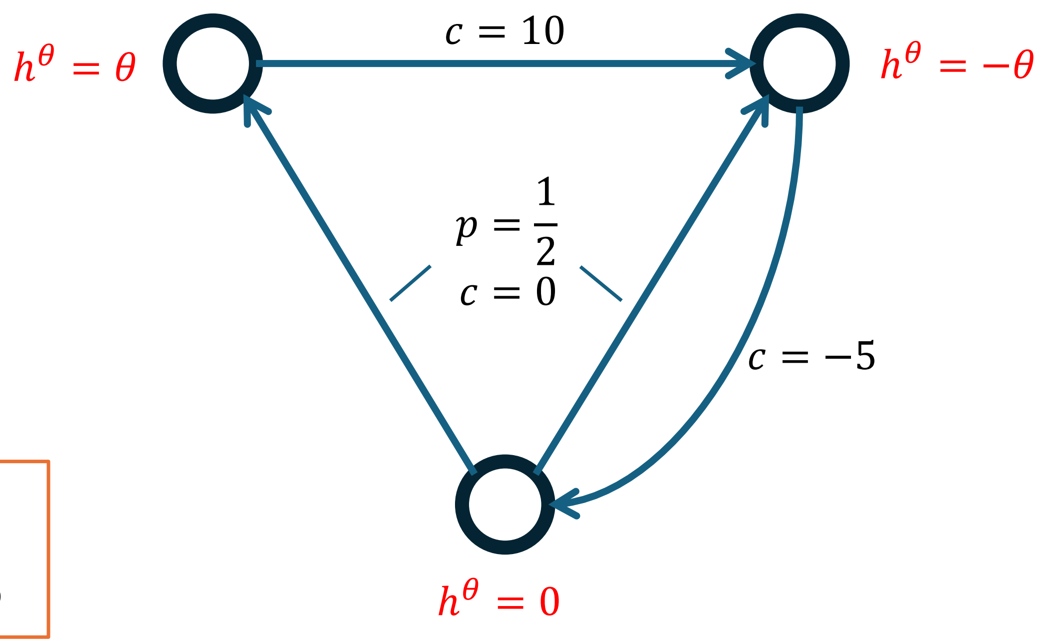
$$\theta_{n+1} = \theta_n + \alpha_{n+1}\mathcal{D}_{n+1}\zeta_{n+1} \tag{9.32}$$

with $\mathcal{D}_{n+1} \stackrel{\text{def}}{=} \mathcal{D}^{\theta_n}_{n+1}$, and

$$\zeta_{n+1} = -\nabla\mathcal{D}^{\theta}_{n+1}\big|_{\theta=\theta_n} = \nabla_\theta[h^\theta(X(n)) - \gamma h^\theta(X(n+1))]\big|_{\theta=\theta_n}$$

Easier to implement, but the MSTD is not always a good metric

Example: MSBE vs MSTD



$$\theta^*_{MSBE} = 5$$
$$\theta^*_{MSTD} = 3{,}75$$

In the above example, θ^*_{MSBE} is optimal since the associated Bellman error is zero. By contrast, θ^*_{MSTD} is not optimal. The reason in this case is that it is biased toward minimizing θ^2, arising from minimizing the temporal difference associated to the edges going from the lower node.

TD(λ)-learning

TD(λ) algorithm

For initialization $\theta_0\,, \zeta_0 \in \mathbb{R}^d$, the sequence of estimates are defined recursively:

$$\begin{aligned} \theta_{n+1} &= \theta_n + \alpha_{n+1}\zeta_n\mathcal{D}_{n+1} \\ \mathcal{D}_{n+1} &= \big(-h^\theta(X(n)) + c(X(n)) + \gamma h^\theta(X(n+1))\big)\Big|_{\theta=\theta_n} \\ \zeta_{n+1} &= \lambda\gamma\zeta_n + \psi(X(n+1))\,. \end{aligned} \tag{9.37}$$

Eligibility vectors: $\zeta_n = \sum_{i=0}^{\infty} (\lambda\gamma)^i \psi(X(n-i))$

Approximation error of TD(λ)-learning

If convergence, then $0 = \mathsf{E}\big[\{-h^{\theta^*}(X(k)) + c(X(k)) + \gamma h^{\theta^*}(X(k+1))\}\zeta_k(i)\big]\,,\quad 1 \le i \le d\,.$

Interpretations for two cases:

If $\lambda = 0$, then $\widehat{\mathbb{E}}_\psi\big[\, D^{\theta^*}_{n+1}(X) \mid X(n) \,\big] = 0$

If $\lambda = 1$, then $\theta^* = \arg\min_\theta \big\| h^\theta - h \big\|_\pi$

See [1, Theorem 9.7].

Convergence of TD(λ)-learning

TD(λ) is a linear recursion:

$$\theta_{n+1} = \theta_n + \alpha_{n+1}\big[A_{n+1}\theta_n - b_{n+1}\big]$$

$$A_{n+1} = \zeta_n\big[\gamma\psi(X(n+1)) - \psi(X(n))\big]^{\intercal}$$
$$b_{n+1} = -\zeta_n c(X(n))$$

Under mild assumptions, $A \stackrel{\text{def}}{=} \mathbb{E}[A_n]$ is Hurwitz

This ensures convergence of the recursion to $\theta^* = A^{-1}b$ under adequate choice of step-sizes $\{\alpha_n\}$, where $b \stackrel{\text{def}}{=} \mathbb{E}[b_n]$

See [1, Theorem 9.8]. See [1, Theorem 8.10] for valid step-size choices.

Least-square TD(λ)-learning

LSTD(λ)

With initialization $\theta_0, \zeta_0 \in \mathbb{R}^d$ and $\widehat{A}_0 \in \mathbb{R}^{d\times d}$:

$$\theta_{n+1} = \theta_n - \alpha_{n+1}\widehat{A}_n^{-1}\zeta_n\mathcal{D}_{n+1} \tag{9.42a}$$

$$\mathcal{D}_{n+1} = c(X(n)) + \big[\gamma\psi(X(n+1)) - \psi(X(n))\big]^\intercal\theta_n \tag{9.42b}$$

$$\zeta_{n+1} = \lambda\gamma\zeta_n + \psi(X(n+1)), \tag{9.42c}$$

$$\widehat{A}_{n+1} = \widehat{A}_n + \alpha_{n+1}[A_{n+1} - \widehat{A}_n] \tag{9.42d}$$

$$A_{n+1} = \zeta_n[\gamma\psi(X(n+1)) - \psi(X(n))]^\intercal \tag{9.42e}$$

It is a Stochastic Newton-Raphson method since $\hat{A}_n$ approximates the Jacobian (A) of $A\theta - b$

Nonlinear parameterized TD(λ)-learning

Suppose $\{h^\theta : \theta \in \mathbb{R}^d\}$ are not linear functions of θ, but are differentiable. A generalization of the foregoing is based on the definition

$$\psi_i(x;\theta) = \frac{\partial}{\partial \theta_i} h^\theta(x) .$$

The temporal difference and eligibility sequence are redefined as follows:

$$\mathcal{D}_{n+1} = c(X(n)) + \gamma h^{\theta_n}(X(n+1)) - h^{\theta_n}(X(n)) \tag{9.43a}$$

$$\zeta_{n+1} = \lambda\gamma\zeta_n + \psi(X(n+1);\theta_n) , \qquad n \ge 0 . \tag{9.43b}$$

If the algorithm is convergent, then the limit θ^* is expected to solve

$$0 = \mathsf{E}\big[\big(c(X(n)) + \gamma h^{\theta_n}(X(n+1)) - h^{\theta_n}(X(n))\big)\zeta_{n+1}^{\theta*}\big] \tag{9.44}$$

where $\zeta_{n+1}^{\theta*} = \lambda\gamma\zeta_n^{\theta*} + \psi(X(n+1);\theta^*)$, $n \ge 0$, and the expectation in (9.44) is taken with respect to the joint stationary process $(\boldsymbol{X}, \zeta^{\theta^*})$. The fixed point equation (9.44) no longer has an interpretation as a Galerkin relaxation when the eligibility vector depends upon the parameter θ.

Return to the Q-function

Goal: evaluate the Q-function $Q(x, u)$ of a given policy $\breve{\phi}(u|x)$

"Data": stationary sequence $\Phi(k) = \big(X(k), U(k)\big)$

On-policy: $\mathbb{P}[\, U(k) = u \mid X(k) = x \,] = \breve{\phi}(u|x)$

Off-policy: $\Phi(k)$ is not related to $\breve{\phi}(u|x)$

Bellman equation for the Q-function

- On-policy method: If U is chosen according to the policy $\check{\phi}$ then

$$Q(\Phi(k)) = c(\Phi(k)) + \gamma \mathsf{E}[Q(\Phi(k+1)) \mid \mathcal{F}_k] \tag{9.49}$$

- Off-policy method: If U is *any* admissible input then the representation must be modified:

$$Q(\Phi(k)) = c(\Phi(k)) + \gamma \mathsf{E}[\underline{Q}(X(k+1)) \mid \mathcal{F}_k] \tag{9.50}$$

where $\underline{Q}(x) = \sum_u Q(x,u)\check{\phi}(u|x)$

TD(λ)-learning for the Q-function (on-policy)

TD(λ) algorithm (on-policy for Q)

For initialization $\theta_0 \,, \zeta_0 \in \mathbb{R}^d$, the sequence of estimates are defined recursively:

$$\begin{aligned} \theta_{n+1} &= \theta_n + \alpha_{n+1} \zeta_n \mathcal{D}_{n+1} \\ \mathcal{D}_{n+1} &= \big(-H^\theta(\Phi(n)) + c_n + \gamma H^\theta(\Phi(n+1)) \big) \Big|_{\theta=\theta_n} \\ \zeta_{n+1} &= \lambda \gamma \zeta_n + \psi_{(n+1)} \,, \qquad \psi_{(n+1)} \stackrel{\text{def}}{=} \psi(\Phi(n+1)) \,, \quad c_n \stackrel{\text{def}}{=} c(\Phi(n)) \end{aligned} \tag{9.51}$$

Analysis of TD(λ)-learning (on-policy)

Same results as for TD(λ)-learning for the value function h since $\Phi(k)$ is an autonomous process

Example:

(i) $\lambda = 0$: *In the notation of* (9.19),

$$\widehat{\mathsf{E}}[\mathcal{D}_{n+1}^{\theta^*} \mid Y_n] = 0\,,$$

with $Y_n = \psi(\Phi(n)) = \psi_{(n)}$ *and* $\mathcal{D}_{n+1}^{\theta^*} = -H^{\theta^*}(\Phi(n)) + c_n + \gamma H^{\theta^*}(\Phi(n+1))$.

(ii) $\lambda = 1$: θ^* *solves*

$$\theta^* = \arg\min_\theta \|H^\theta - Q\|_\varpi^2 \stackrel{\text{def}}{=} \sum_{x \in \mathsf{X},\, u \in \mathsf{U}} (H^\theta(x,u) - Q(x,u))^2 \varpi(x,u)$$

Limitations of TD(λ)-learning (on-policy)

Requires a randomized policy to ensure that A is Hurwtiz and that $\|H^\theta - Q\|_\varpi$ is a good metric

Policies from Policy Improvement are not always randomized

Fix this with an "ϵ-perturbation" of the policy

See also "Gibbs' policy"

See [1, § 9.5.1] for the definition of “Gibbs’ policies”.

TD(λ)-learning for the Q-function (off-policy)

TD(λ) algorithm (off-policy for Q)

For initialization $\theta_0\,,\zeta_0 \in \mathbb{R}^d$, the sequence of estimates are defined recursively:

$$\begin{aligned}
\theta_{n+1} &= \theta_n + \alpha_{n+1}\zeta_n\mathcal{D}_{n+1} \\
\mathcal{D}_{n+1} &= \big(-H^\theta(\Phi(n)) + c_n + \gamma\underline{H}^\theta(X(n+1))\big)\Big|_{\theta=\theta_n} \\
\zeta_{n+1} &= \lambda\gamma\zeta_n + \psi_{(n+1)}\,, \qquad \psi_{(n+1)} \stackrel{\text{def}}{=} \psi(\Phi(n+1))\,, \quad c_n \stackrel{\text{def}}{=} c(\Phi(n))
\end{aligned} \tag{9.53}$$

Analysis of TD(λ)-learning (off-policy)

The results that hold for the value function and the Q-function in the on-policy setting are no longer valid

The matrix A and vector b become

$$A = \mathsf{E}_\pi\big[\zeta_n\big(-\psi(\Phi(n)) + \gamma\underline{\psi}(X(n+1))\big)^\intercal\big]\,, \quad b = -\mathsf{E}_\pi\big[c_n\zeta_n\big]\,, \quad \textit{and} \;\; \underline{\psi}(x) = \sum_u \psi(x,u)\check{\phi}(u \mid x).$$

It is not trivial to show that A is invertible (under some assumptions)

It is not guaranteed that A is Hurwitz

See [1, Proposition 9.12] and the discussion below it.

Q-learning

Goal: approximate the optimal Q-function $Q^{\star}(x,u)$

Galerkin relaxation:

Given a parametrized family $\{H^\theta : \theta \in \mathbb{R}^d\}$, and a sequence of d-dimensional eligibility vectors $\{\zeta_n\}$, the goal is to find a solution θ^* to

$$0 = \bar{f}(\theta^*) = \mathsf{E}[\{-H^\theta(\Phi(n)) + c_n + \gamma \underline{H}^\theta(X(n+1))\}\zeta_n]\Big|_{\theta=\theta^*} \tag{9.71}$$

where $\underline{H}(x) = \min_u H(x,u)$

Q(0)-learning

$$\begin{aligned}\theta_{n+1} &= \theta_n + \alpha_{n+1}\mathcal{D}_{n+1}\zeta_n \\ \mathcal{D}_{n+1} &= -H^n(\Phi(n)) + c_n + \gamma\underline{H}^n(X(n+1)) \\ \zeta_n &= \nabla_\theta\{H^\theta(\Phi(n))\}\big|_{\theta=\theta_n} = \psi_{(n)}\end{aligned} \tag{9.75}$$

The recursion (9.75) for the Q-learning algorithm can be written in a form similar to the linear recursion (8.53b). On denoting $\underline{\psi}_{(n+1)} = \psi(X(n+1), \phi_n(X(n+1)))$, with ϕ_n any H^n-greedy policy,

$$\begin{aligned}\theta_{n+1} &= \theta_n + \alpha_{n+1}\big[A_{n+1}\theta_n - b_{n+1}\big] \\ \textit{with} \quad A_{n+1} &= \psi_{(n)}\{\gamma\underline{\psi}_{(n+1)} - \psi_{(n)}\}^\intercal \\ b_{n+1} &= -c_n\psi_{(n)}\end{aligned} \tag{9.76}$$

This is not a linear SA algorithm since the policy ϕ_n depends upon θ_n.

Tabular Q(0)-learning

Proposition 9.15. *The ODE approximation for the Q-learning algorithm* (9.75) *takes the form* $\frac{d}{dt}\theta_t = \bar{f}^0(\theta_t)$*, with vector field*

$$\bar{f}_i^0(\theta) = \varpi(x^i, u^i)[-H^\theta(x^i, u^i) + c(x^i, u^i) + \sum_{x'} \gamma P_{u^i}(x^i, x')\underline{H}^\theta(x')]$$

For each i*, the function* $\bar{f}_i^0$ *is concave and piecewise linear as a function of* θ*.* □

Tabular Q(0)-learning suffers from the "curse of condition number" when $\varpi(x^i, u^i)$ is small

In tabular Q-learning, the state–input space is finite, i.e., $\mathsf{X} \times \mathsf{U} = \{(x^i, u^i) : 1 \le i \le d\}$, and the template ψ is such that $\psi_i(x,u) = \mathbf{1}_{\{(x^i,u^i)\}}(x,u)$. Hence, $\mathsf{E}[\psi_i(\Phi)] = \varpi(x^i, u^i) \triangleq \mathsf{P}[\Phi(n) = (x^i, u^i)]$ (for any fixed n since we are in steady state).

Tabular Q(0)-learning

(continued)

One way to fix the curse of CN is to use a “gain matrix”:

$$\theta_{n+1} = \theta_n + \frac{1}{n+1} G_n \mathcal{D}_{n+1} \zeta_n \,, \qquad G_n^{-1} = \frac{1}{n+1} \sum_{k=0}^{n} \zeta_k \zeta_k^{\intercal} \tag{9.80}$$

Hence,

Its ODE approximation has vector field with components

$$\bar{f}_i(\theta) = -H^\theta(x^i, u^i) + c(x^i, u^i) + \gamma \sum_{x'} P_{u^i}(x^i, x') \underline{H}^\theta(x') \tag{9.81}$$

We can easily see that the matrix G_n^{-1} is diagonal and satisfies $[G_n^{-1}]_{ii}$ equals the proportion of time the process Φ has been in state-input (x^i, u^i) over the interval $k = 0, \ldots, n$. From this observation, (9.81) follows.

Tabular Q(0)-learning

(continued)

Under some assumptions, the recursion (9.80) converges toward θ^*

Stability:

Proposition 9.17. *For Watkins' algorithm* (9.80),

The function $V(\theta) = \|\tilde{\theta}\|_\infty$ *is a Lyapunov function for the ODE with vector field* (9.81):

$$\tfrac{d^+}{dt} V(\vartheta_t) \le -(1-\gamma) V(\vartheta_t)$$

Variance:

Lemma 9.18. *Suppose that the optimal policy* $\phi^\star$ *is unique. Then the Jacobian* $A = \partial \bar{f}(\theta^*)$, *with* $\bar{f}$ *given in* (9.81), *is given by*

$$A = -I + \gamma T^\star \tag{9.83}$$

where $T^\star$ *defines the transition matrix for* Φ *under the optimal policy:*

$$T^\star(i,j) \stackrel{\text{def}}{=} P_{u^i}(x^i, x^j) \mathbb{1}\{u^j = \phi^\star(x^j)\}\,, \qquad 1 \le i,j \le d$$

Proofs are given in [1].

Limitations of general Q(0)-learning

Outside of the tabular setting, very little is known about the convergence of Q(0)-learning

It is not even clear that $\bar{f}(\theta) = 0$ admits a solution!

One way to fix the existence of solution is GQ-learning (next)

GQ-learning

Goal: solve

$$\min_{\theta} \Gamma(\theta) = \min_{\theta} \frac{1}{2}\ \bar{f}(\theta)^{\top} M \bar{f}(\theta)$$

where $M^{-1} = \mathbb{E}\left[\psi_{(n)}\psi_{(n)}^{\top}\right]$

Gradient descent: $\theta_{n+1} = \theta_n + \alpha_{n+1}\left(-\nabla_{\theta}\bar{f}(\theta_n)^{\top} M \bar{f}(\theta_n)\right)$

where $\nabla_{\theta}\bar{f}(\theta_n) = A(\theta_n) \stackrel{\text{def}}{=} \mathbb{E}\left[\psi_{(n)}\left(-\psi_{(n)} + \gamma\underline{\psi}_{(n+1)}\right)^{\top}\right]$

The expression for $\nabla_\theta \bar{f}(\theta_n)$ supposes ϕ_n (the greedy policy associated to θ_n) piecewise constant with respect to θ_n (which is satisfied for finite state–input systems). In fact, the analysis and motivation of GQ-learning is done here for finite state–input systems, but the same algorithm applies to infinite systems.

GQ-learning stochastic gradient descent

GQ-learning

For initialization $\theta_0\,,\ \omega_0 \in \mathbb{R}^d$,

$$\theta_{n+1} = \theta_n + \alpha_{n+1}\big\{\mathcal{D}_{n+1}\psi_{(n)} - \gamma\omega_{n+1}^{\intercal}\psi_{(n)}\underline{\psi}_{(n+1)}\big\} \tag{9.94a}$$

$$\omega_{n+1} = \omega_n + \beta_{n+1}\psi_{(n)}\big\{\mathcal{D}_{n+1} - \psi_{(n)}^{\intercal}\omega_n\big\} \tag{9.94b}$$

$$\textit{where}\quad \underline{\psi}_{(n+1)} = \psi(X(n+1), \phi_n(X(n+1)))$$
$$\mathcal{D}_{n+1} = -H^n(\Phi(n)) + c_n + \gamma\underline{H}^n(X(n+1))$$

where the two step-size sequences satisfy (8.22).

GQ analysis The fast time scale recursion (9.94b) is designed so that $\omega_n \approx M\bar{f}(\theta_n)$ for large n. Theory for two time-scale SA provides an approximation of (9.94a):

$$\theta_{n+1} \approx \theta_n + \alpha_{n+1}\big\{\mathcal{D}_{n+1}\zeta_n - \gamma\bar{f}(\theta_n)^{\intercal}M\zeta_n\underline{\psi}_{(n+1)}\big\}$$

The equation at the bottom implies that the associated ODE approximation has vector field

$$
\begin{aligned}
\bar{f}_{\rm GQ}(\theta) &= \mathsf{E}\left[\mathcal{D}_{n+1}\zeta_n - \gamma \bar{f}(\theta)^\top M \zeta_n \underline{\psi}_{(n+1)}\right] \\
&= \bar{f}(\theta) - \gamma \mathsf{E}\left[\underline{\psi}_{(n+1)} \psi_{(n)}^\top\right] M \bar{f}(\theta) \\
&= \left\{\mathsf{E}\left[\psi_{(n)}\psi_{(n)}^\top\right] - \gamma \mathsf{E}\left[\underline{\psi}_{(n+1)}\psi_{(n)}^\top\right]\right\} M \bar{f}(\theta) \\
&= -A(\theta)^\top M \bar{f}(\theta).
\end{aligned}
$$

Hence, it is indeed a stochastic gradient descent.

Discussion of GQ-learning

Pros: works even if $\bar{f}(\theta) = 0$ has no solution

Cons: the condition number at θ^* can be high when $\gamma \approx 1$

In the tabular setting, it is expected to be $O\big((1-\gamma)^{-2}\big)$

By comparison, for tabular Q(0)-learning, it is $O\big((1-\gamma)^{-1}\big)$

See [1, Proposition 9.27].

Next course

- Actor-critic methods
  - Find the best policy (actor) with respect to some cost metric (critic)
  - Remove the bias inherent to Bellman error metrics

References

[1] Sean Meyn. *Control systems and reinforcement learning.* Cambridge University Press, 2022.