Reinforcement Learning

Lecture 8: RL algorithms with function approximation

Alexandre Proutiere, Sadegh Talebi, Jungseul Ok

KTH, The Royal Institute of Technology
Objectives of this lecture

Present examples of $Q$-learning like algorithms with function approximation.

- What is function approximation?
- How to modify $Q$-learning algorithms and analyse their convergence?
Lecture 8: Outline

1. RL with function approximation
2. Classical $Q$-learning
3. The case of linear function approximation
1. **RL with function approximation**
2. Classical $Q$-learning
3. The case of linear function approximation
Idea: restrict our attention to $Q$-value functions belonging to a parametrised family of functions $Q$.

Examples:

1. Linear functions: $Q = \{Q_\theta, \theta \in \mathbb{R}^M \}$,

$$Q_\theta(s, a) = \sum_{i=1}^{M} \phi_i(s, a)\theta_i = \phi^\top \theta$$

where the $\phi_i$'s are linearly independent.

2. Deep networks: $Q = \{Q_w, w \in \mathbb{R}^M \}$, $Q_w(s, a)$ given as the output of a neural network with weights $w$ and inputs $(s, a)$.
RL with function approximation

Success stories:

- TD-Gammon (Backgammon), Tesauro 1995 (neural nets)
- Acrobatic helicopter autopilots, Ng et al. 2006
- Jeopardy, IBM Watson, 2011
- 49 atari games, pixel-level visual inputs, Google Deepmind 2015
1. RL with function approximation
2. Classical $Q$-learning
3. The case of linear function approximation
Infinite-horizon discounted MDP

- Stationary transition probabilities: \( p(s' | s, a) \)
- Stationary reward: \( r(s, a) \), uniformly bounded
- Objective: for a given discount factor \( \lambda \in [0, 1) \), find a policy \( \pi \in MD \) maximising (over all possible policies)

\[
\lim_{T \to \infty} \mathbb{E} \left[ \sum_{u=0}^{T} \lambda^u r(s_u^\pi, a_u^\pi) \right]
\]
Discounted RL

- Learn $\pi^*$ (the optimal policy) from the data
- Off-policy design problems. Data = (a given trajectory $(s_t, a_t, r(s_t, a_t))_{t=0}^T$), output at time $T = \pi_T$
- On-policy design problems. In each step $t$:
  - Observe the transition $s_{t-1}, a_{t-1}, s_t$ and the reward $r(s_{t-1}, a_{t-1})$
  - Devise a policy $\pi_t$ and select the next action $a_t = \pi_t(s_t)$

How can we solve Bellman’s fixed point equation w/o knowing $p$ and $r$?

$$V^*(s) = \max_{a \in A_s} (r(s, a) + \lambda \sum_j p(j \mid s, a)V^*(j))$$

$$\pi^*(s) \in \arg \max_{a \in A_s} (r(s, a) + \lambda \sum_j p(j \mid s, a)V^*(j))$$
Stochastic Approximation

Find the root of an increasing function from noisy measurements

Assume that at the $n$-th iteration, you select $x_n$
You get a noisy measurement $y_n = h(x_n) + M_n$ with $\mathbb{E}[M_n] = 0$
A generic SA algorithm

Let \( X_n = (X_n(1), \ldots, X_n(d))^{\top} \in \mathbb{R}^d \) satisfying:

\[
X_{n+1} = X_n + \alpha_n [h(X_n) + M_{n+1} + N_{n+1}],
\]

Assumptions:

(A1) \( h : \mathbb{R}^d \to \mathbb{R}^d \) is lipschitz

(A2) (Diminishing step sizes) \( \sum_{n=1}^{\infty} \alpha_n = \infty \) and \( \sum_{n=1}^{\infty} \alpha_n^2 < \infty \).

(A3) (Martingale di\( r \)erence) \( \forall n, \ E[M_{n+1} | F_n] = 0 \) where

\( F_n = \sigma(X_0, M_1, N_1, \ldots, M_n, N_n, X_n) \) and \( \forall n, \ E[\|M_{n+1}\|^2 | F_n] \leq c_0 (1 + \|X_n\|^2) \).

(A4) (Additional noise) \( \forall n, \ \|N_n\|^2 \leq c_n (1 + \|X_n\|^2) \) a.s., where

\( \lim_{n \to \infty} c_n = 0 \) a.s..

(A5) (Stability) \( \dot{x} = h(x) \) has a unique globally stable equilibrium \( x^* \). \( \forall x, \ h_\infty(x) = \lim_{c \to \infty} \frac{h(cx)}{c} \) exists and 0 is the only globally stable point of \( \dot{x} = h_\infty(x) \).
Convergence

Let \( X_n = (X_n(1), \ldots, X_n(d))^\top \in \mathbb{R}^d \) satisfying for all \( n \):

\[
X_{n+1} = X_n + \alpha_n[h(X_n) + M_{n+1} + N_{n+1}],
\]

**Theorem.** If (A1)-(A5) hold, for any initial condition \( X_0 \),

\[
\lim_{n \to \infty} X_n = x^*, \quad \text{almost surely,}
\]

where \( x^* \) is the only globally stable point of \( \dot{x} = h(x) \).
Asynchornous SA algorithm

Let $X_n = (X_n(1), \ldots, X_n(d))^\top \in \mathbb{R}^d$. At each iteration $n$, only a random set of coordinates $I_n \subset \{1, \ldots, d\}$ of $X_n$ are updated: for $1 \leq i \leq d$,

$$X_{n+1}(i) = \begin{cases} 
X_n(i) + \alpha \mathcal{I}_n(i) \left[ h(X_n; i) + M_{n+1}(i) + N_{n+1}(i) \right] & \text{if } i \in I_n \\
X_n(i) & \text{otherwise}
\end{cases}$$

where $\mathcal{I}_n(i)$ is the number of updates of the $i$-th coordinate up to time $n$, i.e., $\mathcal{I}_n(i) := \sum_{m=0}^{n} 1[i \in I_m]$ and $h(x; i)$ is the $i$-th entry of $h(x)$. 
Asynchornous SA algorithm

Assumptions:

(B1) (Linearily growing $\mathcal{I}_n(i)$) There exists a deterministic $\Delta > 0$ such that for all $1 \leq i \leq d$, $\liminf_{n \to \infty} \mathcal{I}_n(i)/n \geq \Delta$ a.s. Furthermore, for $c > 0$ and all $1 \leq i, j \leq d$, the limit of
\[
\frac{\sum_{m=\mathcal{I}_n(i)}^{\mathcal{I}_n(c,i)} \alpha_m}{\sum_{m=\mathcal{I}_n(j)}^{\mathcal{I}_n(c,j)} \alpha_m}
\] as $n \to \infty$ exists a.s. where $\mathcal{I}_n(c, i) := \mathcal{I}_{N_n(c)}(i)$ with
\[
N_n(c) := \min \left\{ N > n : \sum_{m=n+1}^{N} \alpha_m > c \right\}.
\]

(B2) (Slowly decreasing $\alpha_n$) The sequence $\{\alpha_n\}$ satisfies that $\alpha_{n+1} \leq \alpha_n$ eventually and that for $c \in (0, 1)$, $\sup_n \alpha_{[cn]}/\alpha_n < \infty$ and
\[
\frac{\sum_{m=0}^{[cn]} \alpha_m}{\sum_{m=0}^{n} \alpha_m} \to 1,
\] where $[cn]$ is the integer part of $cn$. 
Let \( X_n = (X_n(1), ..., X_n(d))^\top \in \mathbb{R}^d \) satisfying for all \( n \) and for \( 1 \leq i \leq d \),

\[
X_{n+1}(i) = \begin{cases} 
X_n(i) + \alpha \mathbb{1}_n(i)[h(X_n; i) + M_{n+1}(i) + N_{n+1}(i)] & \text{if } i \in I_n \\
X_n(i) & \text{otherwise}
\end{cases}
\]

**Theorem.** If (A1)-(A5) and (B1)-(B2) hold, for any initial condition \( X_0 \),

\[
\lim_{n \to \infty} X_n = x^*, \quad \text{almost surely,}
\]

where \( x^* \) is the only globally stable point of \( \dot{x} = h(x) \).
We apply the R-M algorithm to estimate the Q-function instead. $Q(s, a)$ is the maximum expected reward starting from state $s$ and taking action $a$:

$$Q(s, a) = r(s, a) + \lambda \sum_j p(j | s, a) V^*(j)$$

Note that $V^*(s) = \max_{a \in A_s} Q(s, a)$, and hence

$$Q(s, a) = r(s, a) + \lambda \sum_j p(j | s, a) \max_b Q(j, b)$$

$Q$ is the fixed point of an operator $H$ (defined on $\mathbb{R}^{S \times A}$)

$$(HQ)(s, a) = r(s, a) + \lambda \sum_j p(j | s, a) \max_b Q(j, b)$$
Q-learning

We observe the trajectory of the system under some behaviour policy $\pi_b$: $(s_n, a_n, r_n)_{n \geq 0}$, where $r_n$ is the reward collected in the $n$-th step.

**Parameter.** Step sizes ($\alpha_n$)

1. **Initialization.** Select a Q-function $Q_0 \in \mathbb{R}^{S \times A}$

2. **Q-function improvement.** For $n \geq 0$. Update the Q-function as follows: $\forall s, a$

$$Q_{n+1}(s_n, a_n) = Q_n(s_n, a_n)$$

$$+ 1_{(s, a) = (s_n, a_n)} \alpha_n(s_n, a_n) \left[ r_n + \gamma \max_{b \in A} Q_n(s_{n+1}, b) - Q_n(s_n, a_n) \right]$$

where $\nu_n(s, a) := \sum_{m=0}^{n} 1[(s, a) = (s_m, a_m)]$. 

**Theorem.** Assume that the step sizes \((\alpha_n)\) satisfy (A2), and that the sets \((I_n)\) defined through the behaviour policy \(\pi_b\) satisfy (B1)-(B2). For any discount factor \(\lambda \in (0, 1)\):

\[
\lim_{n \to \infty} Q_n = Q, \quad \text{almost surely.}
\]

The conditions required in the above theorem are met if \(\alpha_n = \frac{1}{n+1}\) and if the behaviour policy yields an irreducible Markov chain (e.g. unichain model). They are also met for the \(\epsilon\)-greedy policy: it selects an action uniformly at random w.p. \(\epsilon\) and \(a_n \in \arg\max_{a \in A_{s_n}} Q_n(s_n, a)\) w.p. \(1 - \epsilon\).
1. RL with function approximation
2. Classical $Q$-learning
3. The case of linear function approximation
Two algorithms with linear function approximation

A. Gradient Temporal Difference (GTD2) for linear $Q$-approximation with stationary target policy\(^1\)

B. Fitted Policy Iteration algorithm for linear $Q$-approximation \(^2\)

---

\(^1\)Maei et al., "Towards off-policy learning control with function approximation", ICML 2010.

\(^2\)Antos et al., "Learning near-optimal policies with Bellman-residual minimisation based fitted policy iteration", JMLR 2008.
Linear function approximation

We look for the best approximation of the $Q$-function that can be written as:

$$Q_\theta(s, a) = \sum_{j=1}^{m} \theta(j) \Phi_j(s, a) = \theta^\top \Phi(s, a),$$

where the $\Phi_j$'s are $m$ linearly independent functions, $\theta \in \mathbb{R}^m$ (parameter to be learnt), $\Phi = (\Phi_1, \ldots, \Phi_m)$. 
A. Off-policy learning with target stationary policy

Let $\pi$ denote a (known) randomised target stationary policy: when in state $s$, $\pi$ selects $a$ with probability $\pi(s, a)$.

The $Q$-function $Q^\pi$ of $\pi$ is defined as the average reward obtained starting in state $s$ and selecting action $a$ and then applying $\pi$ as a decision rule:

$$Q^\pi(s, a) = r(s, a) + \lambda \sum_{s'} p(s'|s, a) \sum_{b \in \mathcal{A}} \pi(s', b) Q^\pi(s', b),$$

or equivalently,

$$Q^\pi = B^\pi Q^\pi.$$

**Objective:** find the best linear approximation of $Q^\pi$ from data collected under the behaviour policy $\pi_b$ defined by (by abusing notations):

$$\pi_b(s, a) = \mathbb{P}[\pi_b(s) = a].$$
A. Objective function

We wish to design an algorithm that updates our estimate of $\theta$ as we process the data, so that $Q_\theta$ converges to the best linear approximation of $Q^\pi$. What is the best approximation?

**Weighted norm:** Let $\mu(s)$ be the stationary probability of being in state $s$ under $\pi_b$, and let $D$ be the diagonal matrix in $[0, 1]^{S \times A}$ whose diagonal entries are $\pi_b(s, a)\mu(s)$. For any function $F : S \times A \to \mathbb{R}$ define:

$$\|F\|_D = \sqrt{\sum_{s, a} \pi_b(s, a)\mu(s)|F(s, a)|^2} = \sqrt{F^\top DF}.$$  

**Projection:** Let $T$ be the projection onto $Q = \{Q_\theta : \theta \in \mathbb{R}^m\}$ w.r.t. $\| \cdot \|_D$:

$$TF = Q_\theta, \quad \text{where } \theta \in \arg\min_{\theta'} \|F - Q_{\theta'}\|_D.$$
A. Objective function

\[ Q_\theta = \Phi \theta \] where \( \Phi \) is a \((SA) \times m\) matrix whose row vectors are \( \Phi(s, a)^\top \). Then we have:

\[ T = \Phi (\Phi^\top D \Phi)^{-1} \Phi^\top D. \]

**Objective function:** we wish to find \( \theta^* \) such that \( Q_{\theta^*} = TB^\pi Q_{\theta^*} \). This is obtained by minimising the *mean square projected Bellman error* (MSPBE):

\[ \varepsilon(\theta) = \| Q_\theta - TB^\pi Q_\theta \|^2_D. \]

Developing we get:

\[ \varepsilon(\theta) = (\Phi^\top D (B^\pi Q_\theta - Q_\theta))^\top (\Phi^\top D \Phi)^{-1} (\Phi^\top D (B^\pi Q_\theta - Q_\theta)). \]
A. Revisiting the objective function

Assume that $s$ is drawn from $\mu$, that $a$ is selected according to $\pi_b$, that the reward $r$ is collected and that the next state is $s'$. Let $\phi = \Phi(s, a)$ and $\phi' = \sum_a \pi(s', a)\Phi(s', a)$. Let $\delta(\theta) = r + \lambda \theta^T \phi' - \theta^T \phi$.

Then:

$$\mathbb{E}[\phi \phi^T] = \sum_{s,a} \mu(s)\pi_b(s,a)\Phi(s,a)\Phi(s,a)^T = \Phi^T D \Phi,$$

$$\mathbb{E}[\delta(\theta)\phi] = \Phi^T D (B^\pi Q_\theta - Q_\theta).$$

Thus:

$$\varepsilon(\theta) = \mathbb{E}[\delta(\theta)\phi]^T \mathbb{E}[\phi \phi^T]^{-1} \mathbb{E}[\delta(\theta)\phi],$$

$$\nabla \varepsilon(\theta) = -2 \mathbb{E}[(\phi - \lambda \phi')\phi^T] \mathbb{E}[\phi \phi^T]^{-1} \mathbb{E}[\delta(\theta)\phi].$$
A. Gradient Temporal Difference

\( \phi \) and \( \phi' \) can be ”observed”, but the gradient involves the product of various expectations. To perform a stochastic gradient descent, we split the gradient into two terms estimated separately.

\[
\begin{align*}
  w &= E[\phi\phi^\top]^{-1}E[\delta(\theta)\phi] \\
  \frac{-1}{2} \nabla \varepsilon(\theta) &= E[(\phi - \lambda \phi')\phi^\top]w
\end{align*}
\]

Algorithm.
Let \( \phi_t = \Phi(s_t, a_t) \) and \( \phi'_t = \sum_a \pi(s'_t, a)\Phi(s'_t, a) \) (observed along the trajectory).

\[
\begin{align*}
  \theta_{t+1} &= \theta_t + \alpha_t(\phi_t - \lambda \phi'_t)\phi_t^\top w_t \\
  w_{t+1} &= w_t + \beta_t(\delta_t(\theta_t) - \phi_t^\top w_t)\phi_t.
\end{align*}
\]
A. Convergence

**Theorem.** Assume that the step sizes \((\alpha_t)\) satisfy (A2), and \(\beta_t = \eta \alpha_t\). Assume that for all \((s, a)\), \(\pi(s, a) > 0\) and \(\pi_b(s, a) > 0\). Finally assume that \(A = \mathbb{E}[\phi(\phi - \lambda\phi')^\top]\) and \(C = \mathbb{E}[\phi\phi^\top]\) are non-singular. Then almost surely, \(\theta_t\) converges to \(\theta^*\) as \(t \to \infty\).
A. Proof

Using $X_t^\top := \left[ \frac{1}{\sqrt{\eta}} w_t^\top, \theta_t^\top \right]$, the algorithm becomes:

$$X_{t+1} = X_t + \alpha_t \sqrt{\eta} (G_{t+1} X_t + g_{t+1}),$$

where we define

$$G_{t+1} := \begin{bmatrix} -\sqrt{\eta} \phi_t \phi_t^\top & \phi_t (\gamma \phi_t' - \phi_t)^\top \\ (\phi_t - \gamma \phi_t') \phi_t^\top & 0 \end{bmatrix}, \quad \text{and} \quad g_{t+1} := \begin{bmatrix} r_t \phi_t \\ 0 \end{bmatrix}.$$

Let $b := \mathbb{E} [r \phi]$. Define

$$G = \begin{bmatrix} -\sqrt{\eta} C & -A \\ A^\top & 0 \end{bmatrix}, \quad \text{and} \quad g = \begin{bmatrix} b \\ 0 \end{bmatrix}.$$
Introduce $h$ and $M_t$ as follows:

$$h(x) = Gx + g ,$$

$$M_{t+1} = (G_{t+1} - \mathbb{E}[G_{n+1} | \mathcal{F}_1]) X_t + g_{t+1} - \mathbb{E}[g_{t+1} | \mathcal{F}_1] ,$$

$$N_{t+1} = (\mathbb{E}[G_{t+1} | \mathcal{F}_1] - G) X_n + \mathbb{E}[g_{t+1} | \mathcal{F}_1] - g ,$$

so that

$$X_{t+1} = X_t + \alpha'_t (h(X_t) + M_{t+1} + N_{t+1}) ,$$

where $\alpha'_t = \sqrt{\eta} \alpha_t$. 
How can we learn the optimal policy?

Data: a trajectory generated through a behaviour policy $\pi_b$.

One solution: Sequentially go through the data refine the $Q$-function approximation using GTD2 algorithm, by changing the target policy (set it to the greedy policy w.r.t. the current $Q$-function estimate.

Let $Q$ be a function $S \times A \rightarrow \mathbb{R}$, the greedy policy w.r.t. $Q$ is

$$\pi^Q(s) \in \arg\max_a Q(s, a).$$
B. Fitted Policy Iteration

Algorithm. FPI

Input: Data \( D = (s_t, a_t, r_t)_{t=1,...,N} \)

Initialisation: Select a function \( Q \) arbitrarily,

For \( k = 1, \ldots, K \):

1. \( \pi^Q \) greedy w.r.t. \( Q \)
2. \( Q \leftarrow GTD2(D, \pi^Q, \pi_b) \)

Performance guarantee:

\[ \| Q^* - Q^K \|_\infty \leq \text{model-expressivity} + \text{rate-of-convergence} \]
References


