Statistical Data Analysis

Jana de Wiljes

Monte-Carlo Algorithm

Let $(x_0^i = x, x_1^i = x, , \ldots, x_{T_i}^i = x)_{i=1}^n$ be a set of n independent trajectories starting from x and terminating after T_i steps. For any $t \leq T_i$, we denote by

$$\hat{R}^i(x_t^i) = \left[R^\mu(x_t^i) + R^\mu(x_t^i) + \cdots + R^\mu(x_T^i) \right] \tag{1}$$

the return of the i-th trajectory at state x_t^i. Then the Monte-Carlo estimator of $V_\mu(x)$ is

$$V_N(x) = \frac{1}{N} \sum_{i=1}^{N} [R_\mu(x_0^i) + R_\mu(x_1^i) + \cdots + R_\mu(x_T^i)] \quad = \frac{1}{N} \sum_{i=1}^{N} \hat{R}^i(x) \tag{2}$$

Monte-Carlo Algorithm

All the returns are unbiased estimators of $V_\mu(x)$ since

$$\mathbb{E}[\hat{R}^i(x_t^i)] = \mathbb{E}\Big[R^\mu(x_t^i) + R^\mu(x_t^i) + \cdots + R^\mu(x_T^i)\Big] = V_\mu(x) \qquad (3)$$

then

$$V_N(x) \xrightarrow{a.s.} V_\mu(x) \qquad (4)$$

First-visit and Every-Visit Monte-Carlo

Remark: any trajectory $(x_0, x_1, x_2, \ldots, x_T)$ contains also the sub-trajectory $(x_t, x_{t+1}, x_{t+1}, \ldots, x_T)$ whose return

$$\hat{R}(x_t) = R^\mu(x_t) + \cdots + R^\mu(x_{T-1}) \tag{5}$$

could be used to build an estimator of $V^\mu(x_t)$.

- **First-visit MC:** For each state x we only consider the sub-trajectory when x is first achieved. Unbiased estimator, only one sample per trajectory.
  - Unbiased estimator
  - only one sample per trajectory
- **Every-visit MC:** Given a trajectory $(x_0 = x, x_1, x_2, \ldots, x_T)$ we list all the m sub-trajectories starting from x up to xT and we average them all to obtain an estimate.
  - More than one sample per trajectory
  - biased estimator

Trade-off: More samples or no bias? $\implies$ Sometimes a biased estimator is preferable if consistent!

TD(1) Algorithm

Motivation: MC requires all the trajectories to be available at once, can we update the estimator online?

TD(1) Algorithm: Let $(x_0^n = x, x_1^n, x_2^n, \dots, x_{T_n}^n)$ be the n-th trajectory and $\hat{R}^n$ be the n corresponding return. For all x_t with $t \leq T - 1$ observed along the trajectory, we update the value function estimate as

$$V_n(x_t^n) = (1 - \eta_n(x_t^n))V_{n-1}(x_t^n) + \eta_n(x_t^n)\hat{R}^n(x_t^n) \tag{6}$$

TD(1) Algorithm

Each sample is an unbiased estimator of the value function

$$\mathbb{E}\Big[R^\mu(x_t^i) + R^\mu(x_t^i) + \cdots + R^\mu(x_T^i)\Big] = V_\mu(x) \tag{7}$$

then the convergence result of stochastic approximation of a mean applies and if all the states are visited in an infinite number of trajectories and for all $x \in X$

$$\sum_{n\geq 0} \eta_n = \infty, \quad \sum_{n\geq 0} \eta_n^2 < \infty, \tag{8}$$

then for any $x \in X$

$$V_n(x) \xrightarrow{a.s.} V(x) \tag{9}$$

$TD(0)$ **Algorithm**

Idea: stochastic approximation for fixed point

- Noisy observation of the operator $\mathcal{T}^\mu$:

$$\hat{\mathcal{T}}^\mu V(x_t) = R^\mu(x_t) + V(x_{t+1}) \tag{10}$$

with $x_t = x$

- Unbiased estimator of $\mathcal{T}^\mu V(x)$ since

$$\begin{aligned}
\mathbb{E}[\hat{\mathcal{T}}^\mu V(x_t)|x_t = x] &= \mathbb{E}[R^\mu(x_t) + V(X_{t+1}|x_t = x] \qquad (11)\\
&= r(x, \mu(x)) + \sum_y P(y|x, \mu(x))V(y) \qquad (12)\\
&= \mathcal{T}^\mu V(x_t) \qquad (13)
\end{aligned}$$

with $x_t = x$

$TD(0)$ **Algorithm**

TD(0) Algorithm: Let $(x_0^n = x, x_1^n, x_2^n, \dots, x_{T_n}^n)$ be the n-th trajectory and $\{\hat{\mathcal{T}}^\mu V_{n-1}(x_t^n)\}_t$ the noisy observation of the operator $\mathcal{T}^\mu$. For all x_t^n with $t \leq T^n - 1$ we update the value function estimate asvalue function estimate as

$$V_n(x_t^n) = (1 - \eta_n(x_t^n))V_{n-1}(x_t^n) + \eta_n(x_t^n)\hat{\mathcal{T}}^\mu V_{n-1}(x_t^n) \quad (14)$$

$$= (1 - \eta_n(x_t^n))V_{n-1}(x_t^n) + \eta_n(x_t^n)(R^\mu(x_t^n) + V_{n-1}(x_{t+1}^n)) \quad (15)$$

TD(0) Algorithm

Theorem: If all the states are visited in an infinite number of trajectories and for all $x \in X$ and

$$\sum_{n \geq 0} \eta_n = \infty, \quad \sum_{n \geq 0} \eta_n^2 < \infty, \tag{16}$$

then for any $x \in X$

$$V_n(x) \xrightarrow{a.s.} V(x) \tag{17}$$

Temporal difference

Def: At iteration n, given the estimator V_{n-s} and a transition from state x_t to state x_{t+1} we define the temporal difference

$$d_t = (R^{\mu}(x_t^n) + V_{n-1}(x_{t+1})) - V_{n-1}(x_t) \tag{18}$$

Remark: Recalling the definition of Bellman equation for state value function, the temporal difference d_t^n provides a measure of coherence of the estimator $V_{n?1}$ w.r.t. the transition $x_t \rightarrow x_{t+1}$.

$TD(0)$ **Algorithm**

TD(0) Algorithm: Let $(x_0^n = x, x_1^n, x_2^n, \ldots, x_{T_n}^n)$ be the n-th trajectory and $\{\hat{\mathcal{T}}^\mu V_{n-1}(x_t^n)\}_t$ the noisy observation of the operator $\mathcal{T}^\mu$. For all x_t^n with $t \leq T^n - 1$ we update the value function estimate asvalue function estimate as

$$V_n(x_t^n) = V_{n-1}(x_t^n) + \eta_n(x_t^n) d_t^n \tag{19}$$

Sarsa

ϵ-Greedy Exploration

General idea of Exploration

- simple idea for a continuous exploration
- all actions are performed with probability larger than zero

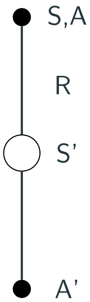
ϵ-Greedy Exploration

- an action is chosen with probability 1-ϵ according to the greedy-policy
- one of the other actions is chosen with probability ϵ (random policy)

$$\mu(a|s) = \begin{cases} a^* = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, Q(s,a) & \text{with probability } 1-\epsilon \\ \text{any action a} & \text{with probability } \frac{\epsilon}{m} \end{cases}$$

where $m = |\mathcal{A}|$

Update of the Action-value function via SARSA



$$Q(S,A) \leftarrow Q(S,A) + \alpha\left(R + \gamma Q(S',A') - Q(S,A)\right)$$

$$Q(S,A) \leftarrow (1-\alpha)Q(S,A) + \alpha\left(R + \gamma Q(S',A')\right)$$

Sarsa

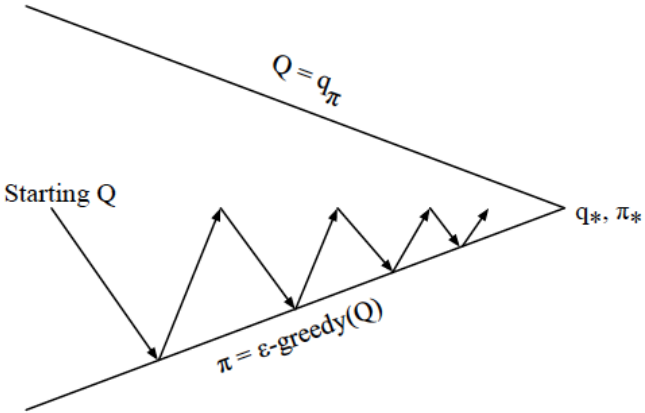


Figure adapted from Silver (2015) Lecture 5: Model-Free Control

In Every time step:

Policy-evaluation: Sarsa, $\hat{Q} \approx Q^{\mu}$

Policy-Optimisation: ϵ-Greedy-Strategie-Optimierung

Sarsa-Algorithm

Initialize $Q(s, a)\ \forall s \in \mathcal{S},\ a \in \mathcal{A}(s)$ randomly

Set: $Q(\text{terminal state}, \cdot) = 0$

Repeat (for each episode ξ):

    Initialize s

    Choose admissible a according to ϵ- greedy policy

    Repeat (for each step of episode):

        Perform action a, observe r, s'

        Choose admissible a' according to ϵ- greedy policy

$$\hat{Q}(s, a) \leftarrow \hat{Q}(s, a) + \alpha \left(R + \gamma \hat{Q}(s', a') - \hat{Q}(s, a) \right)$$

        $s \leftarrow s'$; $a \leftarrow a'$

Q-Learning

Off-Policy idea by means of Q-Learning

Off-Policy-Learning:

- next action is chosen according to the current policy, i.e., $a \sim \mu(\cdot|s)$
- but future actions can be chosen according to an alternative policy e.g.,

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(r + \gamma Q(s', a') - Q(s,a) \right)$$

Off-Policy-Learning:

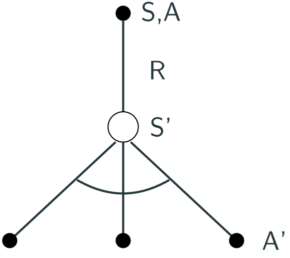
- Idea: want to use

$$\mu(s) = \operatorname*{argmax}_{a'} Q(s, a')$$

- Note

$$\begin{aligned} & r + \gamma Q(s', a') \\ = \; & r + \gamma Q(s', \operatorname*{argmax}_{a'} Q(s', a')) \\ = \; & r + \max_{a'} \gamma Q(s', a') \end{aligned}$$

Q-Learning idea (SARSAMAX)



$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right)$$

Theorem

Q-Learning action value approximation converges to the optimale Action-Value Function, $\hat{Q}(s,a) \rightarrow Q^*(s,a)$.

Q learning-Algorithmus

Initialize $Q(s, a)\ \forall s \in \mathcal{S},\ a \in \mathcal{A}(s)$ randomly
Set: $Q(\text{terminal state}, \cdot) = 0$
Repeat (for each episode ξ):
  Initialize s
  Choose admissible a according to ϵ- greedy policy
  Repeat (for each step of episode):
    Perform action a, observe r, s'
    $\hat{Q}(s, a) \leftarrow \hat{Q}(s, a) + \alpha \left(R + \gamma \max_{a'} \hat{Q}(s', a') - \hat{Q}(s, a) \right)$
    $s \leftarrow s'$; $a \leftarrow a'$