Mahindra™ University
Global Thinkers. Engaged Leaders.

Reinforcement Learning and Autonomous Systems (CS4122)

Lecture 22 (08/10/2024)
Lecture 23 (09/10/2024)
Lecture 24 (14/10/2024)

Instructor: Gourav Saha

Lecture Content

- Temporal Difference (TD) Control.
  - SARSA.
  - Q-Learning.
- On-policy vs Off-policy.

SARSA

- Read section 6.4 of the book. Psuedocode are there in the slides.

SARSA (Original): Psuedocode

(S1): For all $x \in \mathcal{S}$ and all $a \in \mathcal{A}(x)$ arbitrarily initialize q(x, a) to any real value.

(S2): For every episode:

(S3): Reset the episode. This will give the current state x.

(S4): Choose action, a, for current state, x, **using an ε-greedy policy derived from the current Q-function $q(x,\cdot)$.**

(S5): while episode did not end:

(S6): Take action, a. Environment will return reward, r, and go to next state, x'.

(S7): Choose action, a', for the next state, x', **using and ε-greedy policy derived from the current Q-function $q(x',\cdot)$.**

(S8): Update the Q-function only for the current state-action pair (x, a) as follows:

$$q(x,a) = q(x,a) + \alpha\big(r + \beta q(x',a') - q(x,a)\big)$$

(S9): Set $x = x'$ and $a = a'$ for the next time slot.

SARSA (Original): Psuedocode

(S1): For all $x \in \mathcal{S}$ and all $a \in \mathcal{A}(x)$ arbitrarily initialize q(x, a) to any real value.

(S2): For every episode:

(S3): Reset the episode. This will give the current state x.

(S4): Choose action, a, for current state, x, using an ε-greedy policy derived from the current Q-function $Q(x,\cdot)$.

(S5): while episode did not end:

(S6): Take action, a. Environment will return reward, r, and go to next state, x'.

(S7): Choose action, a', for the next state, x', using and ε-greedy policy derived from the current Q-function $q(x',\cdot)$.

(S8): Update the Q-function only for the current state-action pair (x, a) as follows:

$$q(x,a) = Q(x,a) + \alpha\big(r + \beta q(x',a') - q(x,a)\big)$$

(S9): Set $x = x'$ and $a = a'$ for the next time slot.

Learning rate, $\alpha > 0$, and exploration probability, ε, can be time-varying.

By time varying we mean that α and ε can be varied across both episode and time slots in an episode. In practice, no fixed rules; just that they should decrease with time.

SARSA (Variant 1): Psuedocode

(S1): For all $x \in \mathcal{S}$ and all $a \in \mathcal{A}(x)$ arbitrarily initialize q(x, a) to any real value. For all $x \in \mathcal{S}$ arbitrarily initialize V(x) to any real value.

(S2): For every episode:

(S3): Reset the episode. This will give the current state x.

(S4): while episode did not end:

(S5): Choose action, a, for current state, x, using and ε-greedy policy derived from the current Q-function $q(x,\cdot)$.

(S6): Take action, a. Environment will return reward, r, and go to next state, x'.

(S7): Update the Q-function only for the current state-action pair (x, a) as follows:

$$q(x,a) = q(x,a) + \alpha_1\big(r + \beta V(x') - q(x,a)\big)$$

(S8): Update the value function only for the current state x as follows:

$$V(x) = V(x) + \alpha_2\big(r + \beta V(x') - V(x)\big)$$

(S9): Set $x = x'$ and $a = a'$ for the next time slot.

SARSA (Variant 1): Psuedocode

(S1): For all $x \in \mathcal{S}$ and all $a \in \mathcal{A}(x)$ arbitrarily initialize q(x, a) to any real value. For all $x \in \mathcal{S}$ arbitrarily initialize V(x) to any real value.

(S2): For every episode:

(S3): Reset the episode. This will give the current state x.

(S4): while episode did not end:

(S5): Choose action, a, for current state, x, using and ε-greedy policy derived from the current Q-function $q(x,\cdot)$.

(S6): Take action, a. Environment will return reward, r, and go to next state, x'.

(S7): Update the Q-function only for the current state-action pair (x, a) as follows:

$$q(x,a) = q(x,a) + \alpha_1\big(r + \beta V(x') - q(x,a)\big)$$

(S8): Update the value function only for the current state x as follows:

$$V(x) = V(x) + \alpha_2\big(r + \beta V(x') - V(x)\big)$$

Different learning rate is allowed.

(S9): Set $x = x'$ and $a = a'$ for the next time slot.

SARSA (Variant 2)

- Also called **expected-SARSA**.
- Read section 6.6 of the book. Psuedocode are there in the slides.

SARSA (Variant 2): Psuedocode

(S1): For all $x \in \mathcal{S}$ and all $a \in \mathcal{A}(x)$ arbitrarily initialize q(x, a) to any real value. For all $x \in \mathcal{S}$, set $\pi(x)$ to any arbitrary value in $\mathcal{A}(x)$.

(S2): For every episode:

(S3): Reset the episode. This will give the current state x.

(S4): while episode did not end:

(S5): Choose action, a, for current state, x, **using any policy** (preferably ε-greedy policy derived from the current Q-function $q(x, \cdot)$).

(S6): Take action, a. Environment will return reward, r, and go to next state, x'.

(S7): Update the Q-function only for the current state-action pair (x, a) as follows:

$$q(x,a) = q(x,a) + \alpha\left(r + \beta q\left(x', \pi(x')\right) - q(x,a)\right)$$

(S8): Update the policy only for the current state x as follows:

$$\pi(x) = \underset{a \in \mathcal{A}(x)}{\text{argmax}}\ q(x,a)$$

(S9): Set $x = x'$ for the next time slot.

On-Policy vs Off-Policy RL

- There are two kind of RL algorithms: (i) On-policy RL, and (ii) Off-policy RL.

- One challenge with RL (seen this in MC control and SARSA):
  - To ensure good exploration, we have to use an ε-greedy version of the policy that we are trying to learn.
  - In the process of doing so, the Q-function update is based on a modified version of the actual policy.

  Can we have two different policies? On that we are learning (target policy) and the other that is used for generating samples (behavior policy) for the RL agent?

- In on-policy RL, target and behavior policies are the same. E.x. SARSA and its variant 1.

- In off-policy RL, target and behavior policies are different. E.x. Expected SARSA and Q-Learning which we are about to discuss next. **Use of offline data (EXPLAIN).**

Q-Learning

- Read section 6.5 of the book. Psuedocode are there in the slides.
- The intuition behind Q-Learning can't be explained using GPI.

$$
\begin{aligned}
q^{\star}(x,a) &= \overline{r}(x,a) + \beta \sum_{x' \in S} f_T\left[x'|x,a\right] V\left(x'\right) \\
&= \overline{r}(x,a) + \beta \sum_{x' \in S} f_T\left[x'|x,a\right] \max_{a'} q^{\star}\left(x',a'\right)
\end{aligned}
$$

Q-Learning

- Read section 6.5 of the book.
- The intuition behind Q-Learning can't be explained using GPI.

$$
\begin{aligned}
q^{\star}(x,a) &= \overline{r}(x,a) + \beta \sum_{x' \in S} f_T\left[x'|x,a\right] V\left(x'\right) \\
&= \overline{r}(x,a) + \beta \sum_{x' \in S} f_T\left[x'|x,a\right] \max_{a'} q^{\star}\left(x',a'\right) \\
&= \sum_{x' \in S} \overline{r}(x,a)\, f_T\left(x'|x,a\right) + \beta \sum_{x' \in S} f_T\left[x'|x,a\right] \max_{a'} q^{\star}\left(x',a'\right) \\
&= \sum_{x' \in S} \left(\overline{r}(x,a) + \beta \max_{a'} q^{\star}\left(x',a'\right)\right) f_T\left(x'|x,a\right)
\end{aligned}
$$

Q-Learning

- Read section 6.5 of the book.
- The intuition behind Q-Learning can't be explained using GPI.

$$
\begin{aligned}
q^{\star}(x,a) &= \sum_{x^{'} \in S} \left(\overline{r}(x,a) + \beta \max_{a^{'}} q^{\star}\left(x^{'}, a^{'}\right) \right) f_T\left(x^{'}|x,a\right) \\
&= \sum_{x^{'} \in S} \left(\sum_{r} r f_R(r|x,a) + \beta \max_{a^{'}} q^{\star}\left(x^{'}, a^{'}\right) \right) f_T\left(x^{'}|x,a\right) \\
&= \sum_{x^{'} \in S} \left(\sum_{r} r f_R(r|x,a) + \sum_{r} \beta \max_{a^{'}} q^{\star}\left(x^{'}, a^{'}\right) f_R(r|x,a) \right) f_T\left(x^{'}|x,a\right) \\
&= \sum_{x^{'} \in S} \sum_{r} \left(r + \beta \max_{a^{'}} q^{\star}\left(x^{'}, a^{'}\right) \right) f_T\left(x^{'}|x,a\right) f_R(r|x,a)
\end{aligned}
$$

Q-Learning

- Read section 6.5 of the book.
- The intuition behind Q-Learning can't be explained using GPI.

$$q^{\star}(x,a) = E_{x' \sim f_T, r \sim f_R}\left[r + \beta \max_{a'} q^{\star}\left(x', a'\right)\right]$$

Q-Learning: Psuedocode

(S1): For all $x \in \mathcal{S}$ and all $a \in \mathcal{A}(x)$ arbitrarily initialize $\mathrm{q}(x, a)$ to any real value.

(S2): For every episode:

(S3): Reset the episode. This will give the current state x.

(S4): while episode did not end:

(S5): Choose action, a, for current state, x, **using any policy** (preferably ε-greedy policy derived from the current Q-function $q(x,\cdot)$).

(S6): Take action, a. Environment will return reward, r, and go to next state, x'.

(S7): Update the Q-function only for the current state-action pair (x, a) as follows:

$$q(x,a) = q(x,a) + \alpha\left(r + \beta \max_{a' \in \mathcal{A}(x')} q(x',a') - q(x,a)\right)$$

(S8): Set $x = x'$ for the next time slot.


Thank you