Mahindra University

Global Thinkers. Engaged Leaders.

Reinforcement Learning and Autonomous Systems (CS4122)

Lecture 21 (01/10/2024)

Instructor: Gourav Saha

Lecture Content

- Monte Carlo Policy Evaluation.
  - Estimating the Q-function.

- Monte Carlo Control.
  - With exploration restart.
  - ε-greedy version.

Recap and Broad View of this Lecture

$$\pi_{new}(x) = \underset{a \in \mathcal{A}(x)}{\operatorname{argmax}} \; q^{\pi}(x, a)$$

- In the beginning of the previous two lectures we discussed that if we can compute the previous two lectures we discussed that if we can compute the Q-function, $q^{\pi}(x, a)$, corresponding to the current policy, π, then we can use the above equation to improve the policy.

- We also discussed how to compute $V^{\pi}(x)$ for a given policy π. Now computing $V^{\pi}(x)$ is not directly useful as far as learning in MDPs is concerned, the main idea was to start with something simpler to cover the basics. We saw two ways to compute $V^{\pi}(x)$:
  - **Monte Carlo approach.**
  - **Temporal difference (TD) approach.**

  Both these approaches are useful in the wider context of learning in MDPs.

- In this lecture, we will first see how to use Monte Carlo approach to compute $q^{\pi}(x, a)$.

- Immediately after that we will develop two learning algorithms for MDPs using Monte Carlo approach. The fundamental idea behind these algorithms is the above equation and Generalized Policy Iteration (GPI).

MC Policy Evaluation

<u>To compute the Q-function $q^{\pi}(x, a)$</u>

- Read section 5.2 of the book. The title of the section is Monte Carlo Estimation of Action Values.

- IMPORTANT: While reading section 5.2, give specific attention to the paragraph that starts with the following line: "The only complication is that many…". The paragraph after this is also very important.

- Just like computing $V^{\pi}(x)$, we can compute $q^{\pi}(x, a)$ using:
  - First visit Monte Carlo.
  - Every visit Monte Carlo.

  The pseudocode of both these topics is given in the next few slides.

MC Policy Evaluation: Psuedocode for First-Visit MC

Given: A policy, π.

(S1): For all $x \in \mathcal{S}$ and all $a \in \mathcal{A}(x)$ arbitrarily initialize: *(i)* $q(x, a)$ to any real value, and *(ii)* $N(x, a)$ to zero. $q(x, a)$ and $N(x, a)$ are the estimates of the Q-function and the number of samples corresponding to state-action pair (x, a) resp.

(S2): For every episode:

For step (S3), we use the psuedocode to generate a trajectory that we discussed in the previous lecture slides.

(S3): Use policy π to generate a trajectory τ. Let the trajectory be as follows with the last time slot as T:

$$(x_0, a_0, r_0)\,,\ (x_1, a_1, r_1)\,,\ (x_2, a_2, r_2)\,, \ldots,\ (x_T, a_T, r_T)$$

(S4): For all $x \in \mathcal{S}$ and all $a \in \mathcal{A}(x)$, set $visited(x, a)$ to $False$. Set the return corresponding to time $t = 0$ as follows,

$$G_0 = \sum_{t=0}^{T} \beta^t r_t$$

(S5): For $t = 0, 1, 2, \cdots, T$:

(S6): If not$\big(visited(x_t, a_t)\big)$:

(S7): Update $q(x_t, a_t)$ as follows:

$$q(x_t, a_t) = q(x_t, a_t) + \frac{1}{N(x_t, a_t) + 1}\left(G_t - q(x_t, a_t)\right)$$

(S8): Update $N(x_t, a_t) = N(x_t, a_t) + 1$. Also set, $visited(x_t, a_t)$ to True.

(S9): Set $G_{t+1} = (G_t - r_t)/\beta$.

MC Policy Evaluation: Psuedocode for First-Visit MC

Given: A policy, π.

(S1): For all $x \in \mathcal{S}$ and all $a \in \mathcal{A}(x)$ arbitrarily initialize $\mathrm{q}(x, a)$ to any real value. $\mathrm{q}(x, a)$ is the estimate of the Q-function.

(S2): For every episode:

For step (S3), we use the psuedocode to generate a trajectory that we discussed in the previous lecture slides.

(S3): Use policy π to generate a trajectory τ. Let the trajectory be as follows with the last time slot as T:

$$(x_0, a_0, r_0)\,,\ (x_1, a_1, r_1)\,,\ (x_2, a_2, r_2)\,, \ldots,\ (x_T, a_T, r_T)$$

(S4): For all $x \in \mathcal{S}$ and all $a \in \mathcal{A}(x)$, set $visited(x, a)$ to $False$. Set the return corresponding to time $\mathrm{t} = 0$ as follows,

$$G_0 = \sum_{t=0}^{T} \beta^t r_t$$

(S5): For $t = 0, 1, 2, \cdots, T$:

(S6): If $\mathrm{not}\big(visited(x_t, a_t)\big)$:

(S7): Update $q(x_t, a_t)$ as follows:

$$q\left(x_t, a_t\right) = q\left(x_t, a_t\right) + \alpha\left(G_t - q\left(x_t, a_t\right)\right)$$

Gradient descent approach. This is also called the stochastic averaging formula.

(S8): Set $G_{t+1} = (G_t - r_t)/\beta$.

MC Policy Evaluation: Psuedocode for Every-Visit MC

Given: A policy, π.

(S1): For all $x \in \mathcal{S}$ and all $a \in \mathcal{A}(x)$ arbitrarily initialize: *(i)* $\mathrm{q}(x, a)$ to any real value, and *(ii)* $N(x, a)$ to zero. $\mathrm{q}(x, a)$ and $N(x, a)$ are the estimates of the Q-function and the number of samples corresponding to state-action pair (x, a) resp.

(S2): For every episode:

For step (S3), we use the psuedocode to generate a trajectory that we discussed in the previous lecture slides.

(S3): Use policy π to generate a trajectory τ. Let the trajectory be as follows with the last time slot as T:

$$(x_0, a_0, r_0)\,,\ (x_1, a_1, r_1)\,,\ (x_2, a_2, r_2)\,, \ldots,\ (x_T, a_T, r_T)$$

(S4): Set the return corresponding to time $\mathrm{t} = 0$ as follows,

$$G_0 = \sum_{t=0}^{T} \beta^t r_t$$

(S5): For $t = 0, 1, 2, \cdots, T$:

(S6): Update $q(x_t, a_t)$ as follows:

$$q\left(x_t, a_t\right) = q\left(x_t, a_t\right) + \frac{1}{N\left(x_t, a_t\right) + 1}\left(G_t - q\left(x_t, a_t\right)\right)$$

(S7): Update $N(x_t, a_t) = N(x_t, a_t) + 1$.

(S8): Set $G_{t+1} = (G_t - r_t)/\beta$.

MC Policy Evaluation: Psuedocode for Every-Visit MC

Given: A policy, π.

(S1): For all $x \in \mathcal{S}$ and all $a \in \mathcal{A}(x)$ arbitrarily initialize $\mathrm{q}(x, a)$ to any real value. $\mathrm{q}(x, a)$ is the estimate of the Q-function.

(S2): For every episode:

For step (S3), we use the psuedocode to generate a trajectory that we discussed in the previous lecture slides.

(S3): Use policy π to generate a trajectory τ. Let the trajectory be as follows with the last time slot as T:

$$(x_0, a_0, r_0), (x_1, a_1, r_1), (x_2, a_2, r_2), \ldots, (x_T, a_T, r_T)$$

(S4): Set the return corresponding to time $\mathrm{t} = 0$ as follows,

$$G_0 = \sum_{t=0}^{T} \beta^t r_t$$

(S5): For $t = 0, 1, 2, \cdots, T$:

(S6): Update $q(x_t, a_t)$ as follows:

$$q(x_t, a_t) = q(x_t, a_t) + \alpha (G_t - q(x_t, a_t))$$

Gradient descent approach. This is also called the stochastic averaging formula.

(S7): Set $G_{t+1} = (G_t - r_t)/\beta$.

MC Control: With Exploration Start

- Monte Carlo Control with Exploration Start is the first algorithm that you are going to learn as far as learning in MDPs go.
- Read section 5.3 of the book.
- The pseudocode of this algorithm is given in the next three slides.

MC Control with ES: Psuedocode

(S1): For all $x \in \mathcal{S}$ and all $a \in \mathcal{A}(x)$ arbitrarily initialize: *(i)* $q(x,a)$ to any real value, and *(ii)* $N(x,a)$ to zero. $q(x,a)$ and $N(x,a)$ are the estimates of the **optimal** Q-function and the number of samples corresponding to state-action pair (x,a) resp. For all $x \in \mathcal{S}$ arbitrarily initialize a policy $\pi(x)$ to any value in $\mathcal{A}(x)$.

(S2): For every episode: The pseudocode to generate the trajectory in step (S3) is given in **next to next slide**.

(S3): Use policy π to generate a trajectory τ. Let the trajectory be as follows with the last time slot as T:

$$(x_0, a_0, r_0)\,,\ (x_1, a_1, r_1)\,,\ (x_2, a_2, r_2)\,, \ldots,\ (x_T, a_T, r_T)$$

(S4): Set the return corresponding to time $t = 0$ as follows,

$$G_0 = \sum_{t=0}^{T} \beta^t r_t$$

(S5): For $t = 0, 1, 2, \cdots, T$:

(S6): Update $q(x_t, a_t)$ as follows:

$$q(x_t, a_t) = q(x_t, a_t) + \frac{1}{N(x_t, a_t) + 1}\left(G_t - q(x_t, a_t)\right)$$

(S7): Update $N(x_t, a_t) = N(x_t, a_t) + 1$.

(S8): Update $\pi(x_t) = \underset{a \in \mathcal{A}(x_t)}{\operatorname{argmax}}\ q(x_t, a)$.

(S9): Set $G_{t+1} = (G_t - r_t)/\beta$.

MC Control with ES: 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. $\mathrm{q}(x, a)$ is the estimate of the **optimal** Q-function. For all $x \in \mathcal{S}$ arbitrarily initialize a policy $\pi(x)$ to any value in $\mathcal{A}(x)$.

(S2): For every episode: The pseudocode to generate the trajectory in step (S3) is given in **next slide**.

(S3): Use policy π to generate a trajectory τ. Let the trajectory be as follows with the last time slot as T:

$$(x_0, a_0, r_0)\,,\ (x_1, a_1, r_1)\,,\ (x_2, a_2, r_2)\,, \ldots,\ (x_T, a_T, r_T)$$

(S4): Set the return corresponding to time $\mathrm{t} = 0$ as follows,

$$G_0 = \sum_{t=0}^{T} \beta^t r_t$$

(S5): For $t = 0, 1, 2, \cdots, T$:

(S6): Update $q(x_t, a_t)$ as follows:

$$q(x_t, a_t) = q(x_t, a_t) + \alpha\left(G_t - q(x_t, a_t)\right)$$

Gradient descent approach. This is also called the stochastic averaging formula.

(S7): Update $\pi(x_t) = \underset{a \in \mathcal{A}(x_t)}{\operatorname{argmax}}\ q(x_t, a)$.

(S8): Set $G_{t+1} = (G_t - r_t)/\beta$.

MC Control with ES: Psuedocode to Generate Trajectory

Given: A policy, π.

(S1): Reset the environment to initial state x_0 **such that all the state is the state space $\boldsymbol{\mathcal{S}}$ has non-zero probability of getting chosen.**. Initialize time $t = 0$, and an empty list τ that will contain the trajectory for the current episode.

(S2): while episode did not end:

(S3): **If $\boldsymbol{t = 0}$:**

(S4): **Choose action $\boldsymbol{a_0}$ such that all actions in the action space $\boldsymbol{\mathcal{A}(s)}$ has non-zero probability of getting chosen.**

(S5): Else:

(S6): Use policy π for the current state x_t to choose action a_t.

(S7): Take action a_t. Environment will return reward r_t and transition to next state x_{t+1}.

(S8): Append the state, action, reward pair (x_t, a_t, r_t) to τ. Set $t = t + 1$.

(S8): Return trajectory τ.

MC Control ε-greedy policy

- There are two disadvantage of MC Control with Exploration start (descending order of importance):
  - There is not enough exploration because exploration happens only in the beginning of an episode.
  - It is only applicable when we are simulating the environment in our computer because the only can we restart the episode to any state x with non-zero probability. When we are dealing with real environment, there is no control over the initial state (the nature decides it).

- This is where MC control ε-greedy policy comes into picture. It is exactly the MDP equivalent of the ε-greedy policy that we saw for bandit setups.

- Read section 5.4 of the book. You may choose to leave the mathematical analysis starting from page 101.

- The pseudocode for MC control ε-greedy policy is given in the next three slides.

MC Control ε greedy policy: Psuedocode

(S1): For all $x \in \mathcal{S}$ and all $a \in \mathcal{A}(x)$ arbitrarily initialize: *(i)* q(x, a) to any real value, and *(ii)* $N(x, a)$ to zero. q(x, a) and $N(x, a)$ are the estimates of the **optimal** Q-function and the number of samples corresponding to state-action pair (x, a) resp. For all $x \in \mathcal{S}$ arbitrarily initialize a policy $\pi(x)$ to any value in $\mathcal{A}(x)$.

(S2): For every episode: The pseudocode to generate the trajectory in step (S3) is given in **next to next slide**.

(S3): Use policy π to generate a trajectory τ. Let the trajectory be as follows with the last time slot as T:

$$(x_0, a_0, r_0)\,,\ (x_1, a_1, r_1)\,,\ (x_2, a_2, r_2)\,, \ldots,\ (x_T, a_T, r_T)$$

(S4): Set the return corresponding to time $\mathrm{t} = 0$ as follows,

$$G_0 = \sum_{t=0}^{T} \beta^t r_t$$

(S5): For $t = 0, 1, 2, \cdots, T$:

(S6): Update $q(x_t, a_t)$ as follows:

$$q(x_t, a_t) = q(x_t, a_t) + \frac{1}{N(x_t, a_t) + 1} \left(G_t - q(x_t, a_t)\right)$$

(S7): Update $N(x_t, a_t) = N(x_t, a_t) + 1$.

(S8): Set $G_{t+1} = (G_t - r_t)/\beta$.

MC Control ε greedy policy : Psuedocode

(S1): For all $x \in \mathcal{S}$ and all $a \in \mathcal{A}(x)$ arbitrarily initialize $q(x, a)$ to any real value. $q(x, a)$ is the estimate of the **optimal** Q-function. For all $x \in \mathcal{S}$ arbitrarily initialize a policy $\pi(x)$ to any value in $\mathcal{A}(x)$.

(S2): For every episode: The pseudocode to generate the trajectory in step (S3) is given in **next slide**.

(S3): Use policy π to generate a trajectory τ. Let the trajectory be as follows with the last time slot as T:

$$(x_0, a_0, r_0)\,,\ (x_1, a_1, r_1)\,,\ (x_2, a_2, r_2)\,, \ldots,\ (x_T, a_T, r_T)$$

(S4): Set the return corresponding to time $t = 0$ as follows,

$$G_0 = \sum_{t=0}^{T} \beta^t r_t$$

(S5): For $t = 0, 1, 2, \cdots, T$:

(S6): Update $q(x_t, a_t)$ as follows:

$$q(x_t, a_t) = q(x_t, a_t) + \alpha\left(G_t - q(x_t, a_t)\right)$$

Gradient descent approach. This is also called the stochastic averaging formula.

(S7): Set $G_{t+1} = (G_t - r_t)/\beta$.

MC Control ε greedy policy : Psuedocode to Generate Trajectory

Given: **Q-function q($\boldsymbol{x}, \boldsymbol{a}$)** for all $x \in \mathcal{S}$ and all $a \in \mathcal{A}(x)$.

(S1): Reset the environment to get the initial state x_0. Initialize time $t = 0$, and an empty list τ that will contain the trajectory for the current episode.

(S2): while episode did not end:

(S3): Choose action a_t as follows:

(a) Sample a random variable v between 0 to 1 from a uniform distribution.

(b) If $v \leq \varepsilon$, a_t is chosen uniformly at random from $\mathcal{A}(x)$. Else, choose $a_t = \underset{a \in \mathcal{A}(x_t)}{\operatorname{argmax}}\ q(x_t, a)$.

(S4): Take action a_t. Environment will return reward r_t and transition to next state x_{t+1}.

(S5): Append the state, action, reward pair (x_t, a_t, r_t) to τ. Set $t = t + 1$.

(S6): Return trajectory τ.

Thank you