Univ Gustave Eiffel - Cosys / Grettia

Reinforcement Learning and Optimal Control - Master 2 SIA

Monte Carlo Methods

Nadir Farhi
chargé de recherche, UGE - Ifsttar/Cosys/Grettia
nadir.farhi@univ-eiffel.fr

Uni Eiffel - 14 october 2024

Monte Carlo methods

- Monte Carlo methods require only experience, i.e. sample sequences of states, actions, and rewards,

Monte Carlo methods

- Monte Carlo methods require only experience, i.e. sample sequences of states, actions, and rewards,
- from actual or simulated interaction with an environment.

Monte Carlo methods

- Monte Carlo methods require only experience, i.e. sample sequences of states, actions, and rewards,
- from actual or simulated interaction with an environment.
- We do not assume complete knowledge of the environment.

Monte Carlo methods

- Monte Carlo methods require only experience, i.e. sample sequences of states, actions, and rewards,
- from actual or simulated interaction with an environment.
- We do not assume complete knowledge of the environment.
- A model is required but only to generate sample transitions (no need of the complete probability distributions as in DP).

Monte Carlo methods

- Monte Carlo methods require only experience, i.e. sample sequences of states, actions, and rewards,
- from actual or simulated interaction with an environment.
- We do not assume complete knowledge of the environment.
- A model is required but only to generate sample transitions (no need of the complete probability distributions as in DP).
- We define here Monte Carlo methods only for episodic tasks.

Monte Carlo methods

- Monte Carlo methods require only experience, i.e. sample sequences of states, actions, and rewards,
- from actual or simulated interaction with an environment.
- We do not assume complete knowledge of the environment.
- A model is required but only to generate sample transitions (no need of the complete probability distributions as in DP).
- We define here Monte Carlo methods only for episodic tasks.
- Only on the completion of an episode are value estimates and policies changed.

Monte Carlo methods

- Monte Carlo methods require only experience, i.e. sample sequences of states, actions, and rewards,
- from actual or simulated interaction with an environment.
- We do not assume complete knowledge of the environment.
- A model is required but only to generate sample transitions (no need of the complete probability distributions as in DP).
- We define here Monte Carlo methods only for episodic tasks.
- Only on the completion of an episode are value estimates and policies changed.
- Monte Carlo methods can thus be incremental in an episode-by-episode sense, but not in a step-by-step (online) sense.

Monte Carlo methods

- Monte Carlo methods require only experience, i.e. sample sequences of states, actions, and rewards,
- from actual or simulated interaction with an environment.
- We do not assume complete knowledge of the environment.
- A model is required but only to generate sample transitions (no need of the complete probability distributions as in DP).
- We define here Monte Carlo methods only for episodic tasks.
- Only on the completion of an episode are value estimates and policies changed.
- Monte Carlo methods can thus be incremental in an episode-by-episode sense, but not in a step-by-step (online) sense.
- Here we learn value functions from sample returns with the MDP.

Monte Carlo methods

- Monte Carlo methods require only experience, i.e. sample sequences of states, actions, and rewards,
- from actual or simulated interaction with an environment.
- We do not assume complete knowledge of the environment.
- A model is required but only to generate sample transitions (no need of the complete probability distributions as in DP).
- We define here Monte Carlo methods only for episodic tasks.
- Only on the completion of an episode are value estimates and policies changed.
- Monte Carlo methods can thus be incremental in an episode-by-episode sense, but not in a step-by-step (online) sense.
- Here we learn value functions from sample returns with the MDP.
- As in DP, we will see Policy evaluation, policy improvement, and then Policy iterations and GPI.

Monte Carlo Prediction (policy evaluation)

- We learn the state-value function for a given policy.

Monte Carlo Prediction (policy evaluation)

- We learn the state-value function for a given policy.
- Recall : the value of a state is the expected return starting from that state.

Monte Carlo Prediction (policy evaluation)

- We learn the state-value function for a given policy.
- Recall : the value of a state is the expected return starting from that state.
- One way to estimate the state-value function of a state from experience : Average the returns observed after visits to that state.

Monte Carlo Prediction (policy evaluation)

- We learn the state-value function for a given policy.
- Recall : the value of a state is the expected return starting from that state.
- One way to estimate the state-value function of a state from experience : Average the returns observed after visits to that state.
- The average should converge to the expected value.

Monte Carlo Prediction (policy evaluation)

- We learn the state-value function for a given policy.
- Recall : the value of a state is the expected return starting from that state.
- One way to estimate the state-value function of a state from experience : Average the returns observed after visits to that state.
- The average should converge to the expected value.
- First-visit MC method : estimates $v_\pi(s)$ as the average of the returns following first visits to s.

Monte Carlo Prediction (policy evaluation)

- We learn the state-value function for a given policy.
- Recall : the value of a state is the expected return starting from that state.
- One way to estimate the state-value function of a state from experience : Average the returns observed after visits to that state.
- The average should converge to the expected value.
- First-visit MC method : estimates $v_\pi(s)$ as the average of the returns following first visits to s.
- Every-visit MC method : averages the returns following all visits to s.

First-visit MC method

First-visit MC prediction, for estimating $V \approx v_\pi$

Input: a policy π to be evaluated

Initialize:

$V(s) \in \mathbb{R}$, arbitrarily, for all $s \in \mathcal{S}$

$Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Loop forever (for each episode):

Generate an episode following π: $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$

$G \leftarrow 0$

Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:

$G \leftarrow \gamma G + R_{t+1}$

Unless S_t appears in $S_0, S_1, \ldots, S_{t-1}$:

Append G to $Returns(S_t)$

$V(S_t) \leftarrow \text{average}(Returns(S_t))$

First-visit MC method

- DP : All of the probabilities must be computed before DP can be applied.

First-visit MC method

- DP : All of the probabilities must be computed before DP can be applied.
- Such computations are often complex and error-prone.

First-visit MC method

- DP : All of the probabilities must be computed before DP can be applied.
- Such computations are often complex and error-prone.
- First-Visit : generating the sample transitions required by Monte Carlo methods can be easy.

First-visit MC method

- DP : All of the probabilities must be computed before DP can be applied.
- Such computations are often complex and error-prone.
- First-Visit : generating the sample transitions required by Monte Carlo methods can be easy.
- So even when one has complete knowledge of the environment's dynamics, the ability of Monte Carlo methods to work with sample episodes alone can be a significant advantage.

First-visit MC method

- DP : All of the probabilities must be computed before DP can be applied.
- Such computations are often complex and error-prone.
- First-Visit : generating the sample transitions required by Monte Carlo methods can be easy.
- So even when one has complete knowledge of the environment's dynamics, the ability of Monte Carlo methods to work with sample episodes alone can be a significant advantage.
- Unlike DP, in Monte Carlo methods, the estimates for each state are independent, i.e. the estimate for one state does not build upon the estimate of any other state.

First-visit MC method

- DP : All of the probabilities must be computed before DP can be applied.
- Such computations are often complex and error-prone.
- First-Visit : generating the sample transitions required by Monte Carlo methods can be easy.
- So even when one has complete knowledge of the environment's dynamics, the ability of Monte Carlo methods to work with sample episodes alone can be a significant advantage.
- Unlike DP, in Monte Carlo methods, the estimates for each state are independent, i.e. the estimate for one state does not build upon the estimate of any other state.
- The computational expense of estimating the value of a single state is independent of the number of states.

First-visit MC method

- DP : All of the probabilities must be computed before DP can be applied.
- Such computations are often complex and error-prone.
- First-Visit : generating the sample transitions required by Monte Carlo methods can be easy.
- So even when one has complete knowledge of the environment's dynamics, the ability of Monte Carlo methods to work with sample episodes alone can be a significant advantage.
- Unlike DP, in Monte Carlo methods, the estimates for each state are independent, i.e. the estimate for one state does not build upon the estimate of any other state.
- The computational expense of estimating the value of a single state is independent of the number of states.
- This can make Monte Carlo methods particularly attractive when one requires the value of only one or a subset of states.

Monte Carlo Estimation of Action Values

- We consider the policy evaluation problem for action values (estiamtion of $q_\pi(s, a)$.
- We are interested here in visits to a state-action pair rather than to a state.
- A state-action pair (s, a) is said to be visited in an episode if ever the state s is visited and action a is taken in it.
- We assume that the episodes start in a state-action pair, and that every pair has a nonzero probability of being selected as the start.
- This guarantees that all state-action pairs will be visited an infinite number of times in the limit of an infinite number of episodes.
- This is called the assumption of exploring starts.
- The most common alternative approach to assuring that all state-action pairs are encountered is to consider only policies that are stochastic with a nonzero probability of selecting all actions in each state.

Monte Carlo Estimation of Action Values

- We consider the policy evaluation problem for action values (estiamtion of $q_\pi(s, a)$.
- We are interested here in visits to a state-action pair rather than to a state.
- A state-action pair (s, a) is said to be visited in an episode if ever the state s is visited and action a is taken in it.
- We assume that the episodes start in a state-action pair, and that every pair has a nonzero probability of being selected as the start.
- This guarantees that all state-action pairs will be visited an infinite number of times in the limit of an infinite number of episodes.
- This is called the assumption of exploring starts.
- The most common alternative approach to assuring that all state-action pairs are encountered is to consider only policies that are stochastic with a nonzero probability of selecting all actions in each state.

Monte Carlo Estimation of Action Values

- We consider the policy evaluation problem for action values (estiamtion of $q_\pi(s, a)$.
- We are interested here in visits to a state-action pair rather than to a state.
- A state-action pair (s, a) is said to be visited in an episode if ever the state s is visited and action a is taken in it.
- We assume that the episodes start in a state-action pair, and that every pair has a nonzero probability of being selected as the start.
- This guarantees that all state-action pairs will be visited an infinite number of times in the limit of an infinite number of episodes.
- This is called the assumption of exploring starts.
- The most common alternative approach to assuring that all state-action pairs are encountered is to consider only policies that are stochastic with a nonzero probability of selecting all actions in each state.

Monte Carlo Estimation of Action Values

- We consider the policy evaluation problem for action values (estiamtion of $q_\pi(s, a)$.
- We are interested here in visits to a state-action pair rather than to a state.
- A state-action pair (s, a) is said to be visited in an episode if ever the state s is visited and action a is taken in it.
- We assume that the episodes start in a state-action pair, and that every pair has a nonzero probability of being selected as the start.
- This guarantees that all state-action pairs will be visited an infinite number of times in the limit of an infinite number of episodes.
- This is called the assumption of exploring starts.
- The most common alternative approach to assuring that all state-action pairs are encountered is to consider only policies that are stochastic with a nonzero probability of selecting all actions in each state.

Monte Carlo Estimation of Action Values

- We consider the policy evaluation problem for action values (estiamtion of $q_\pi(s, a)$.
- We are interested here in visits to a state-action pair rather than to a state.
- A state-action pair (s, a) is said to be visited in an episode if ever the state s is visited and action a is taken in it.
- We assume that the episodes start in a state-action pair, and that every pair has a nonzero probability of being selected as the start.
- This guarantees that all state-action pairs will be visited an infinite number of times in the limit of an infinite number of episodes.
- This is called the assumption of exploring starts.
- The most common alternative approach to assuring that all state-action pairs are encountered is to consider only policies that are stochastic with a nonzero probability of selecting all actions in each state.

Monte Carlo Estimation of Action Values

- We consider the policy evaluation problem for action values (estiamtion of $q_\pi(s, a)$.
- We are interested here in visits to a state-action pair rather than to a state.
- A state-action pair (s, a) is said to be visited in an episode if ever the state s is visited and action a is taken in it.
- We assume that the episodes start in a state-action pair, and that every pair has a nonzero probability of being selected as the start.
- This guarantees that all state-action pairs will be visited an infinite number of times in the limit of an infinite number of episodes.
- This is called the assumption of exploring starts.
- The most common alternative approach to assuring that all state-action pairs are encountered is to consider only policies that are stochastic with a nonzero probability of selecting all actions in each state.

Monte Carlo Estimation of Action Values

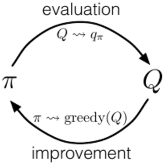
- We consider the policy evaluation problem for action values (estiamtion of $q_\pi(s, a)$.
- We are interested here in visits to a state-action pair rather than to a state.
- A state-action pair (s, a) is said to be visited in an episode if ever the state s is visited and action a is taken in it.
- We assume that the episodes start in a state-action pair, and that every pair has a nonzero probability of being selected as the start.
- This guarantees that all state-action pairs will be visited an infinite number of times in the limit of an infinite number of episodes.
- This is called the assumption of exploring starts.
- The most common alternative approach to assuring that all state-action pairs are encountered is to consider only policies that are stochastic with a nonzero probability of selecting all actions in each state.

Monte Carlo Control

- We consider here how Monte Carlo estimation can be used in control, that is, to approximate optimal policies.

Monte Carlo Control

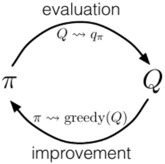
- We consider here how Monte Carlo estimation can be used in control, that is, to approximate optimal policies.
- We follow the idea of the GPI, alternation policy evaluation and policy improvement.



$$\pi_0 \xrightarrow{\text{E}} q_{\pi_0} \xrightarrow{\text{I}} \pi_1 \xrightarrow{\text{E}} q_{\pi_1} \xrightarrow{\text{I}} \pi_2 \xrightarrow{\text{E}} \cdots \xrightarrow{\text{I}} \pi_* \xrightarrow{\text{E}} q_*,$$

Monte Carlo Control

- We consider here how Monte Carlo estimation can be used in control, that is, to approximate optimal policies.
- We follow the idea of the GPI, alternation policy evaluation and policy improvement.

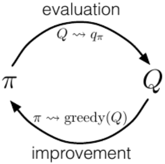


$$\pi_0 \xrightarrow{\text{E}} q_{\pi_0} \xrightarrow{\text{I}} \pi_1 \xrightarrow{\text{E}} q_{\pi_1} \xrightarrow{\text{I}} \pi_2 \xrightarrow{\text{E}} \cdots \xrightarrow{\text{I}} \pi_* \xrightarrow{\text{E}} q_*,$$

- We start with an arbitrary policy π_0 and ending with the optimal policy and optimal action-value function.

Monte Carlo Control

- We consider here how Monte Carlo estimation can be used in control, that is, to approximate optimal policies.
- We follow the idea of the GPI, alternation policy evaluation and policy improvement.

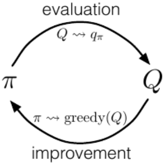


$$\pi_0 \xrightarrow{\mathrm{E}} q_{\pi_0} \xrightarrow{\mathrm{I}} \pi_1 \xrightarrow{\mathrm{E}} q_{\pi_1} \xrightarrow{\mathrm{I}} \pi_2 \xrightarrow{\mathrm{E}} \cdots \xrightarrow{\mathrm{I}} \pi_* \xrightarrow{\mathrm{E}} q_*,$$

- We start with an arbitrary policy π_0 and ending with the optimal policy and optimal action-value function.
- Policy evaluation is done as descriobed above : many episodes are experienced, with the approximate action-value function approaching the true function asymptotically.

Monte Carlo Control

- We consider here how Monte Carlo estimation can be used in control, that is, to approximate optimal policies.
- We follow the idea of the GPI, alternation policy evaluation and policy improvement.



$$\pi_0 \xrightarrow{\text{E}} q_{\pi_0} \xrightarrow{\text{I}} \pi_1 \xrightarrow{\text{E}} q_{\pi_1} \xrightarrow{\text{I}} \pi_2 \xrightarrow{\text{E}} \cdots \xrightarrow{\text{I}} \pi_* \xrightarrow{\text{E}} q_*,$$

- We start with an arbitrary policy π_0 and ending with the optimal policy and optimal action-value function.
- Policy evaluation is done as descriobed above : many episodes are experienced, with the approximate action-value function approaching the true function asymptotically.
- Policy improvement is done by making the policy greedy with respect to the current value function.

$$\pi(s) \doteq \arg\max_a q(s,a).$$

Monte Carlo Exploring Starts

Monte Carlo ES (Exploring Starts), for estimating $\pi \approx \pi_*$

Initialize:

$\pi(s) \in \mathcal{A}(s)$ (arbitrarily), for all $s \in \mathcal{S}$

$Q(s,a) \in \mathbb{R}$ (arbitrarily), for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

$Returns(s,a) \leftarrow$ empty list, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

Loop forever (for each episode):

Choose $S_0 \in \mathcal{S}$, $A_0 \in \mathcal{A}(S_0)$ randomly such that all pairs have probability > 0

Generate an episode from S_0, A_0, following π: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$

$G \leftarrow 0$

Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:

$G \leftarrow \gamma G + R_{t+1}$

Unless the pair S_t, A_t appears in $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:

Append G to $Returns(S_t, A_t)$

$Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$

$\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$

Monte Carlo Control without Exploring Starts

- ε-soft policy : $\pi(a \mid s) \geq \frac{\varepsilon}{|\mathcal{A}(s)|}, \forall s, \forall a.$

Monte Carlo Control without Exploring Starts

- ε-soft policy : $\pi(a \mid s) \geq \frac{\varepsilon}{|\mathcal{A}(s)|}, \forall s, \forall a$.
- Among ε-soft policies, ε-greedy policies are in some sense those that are closest to greedy.
- Indeed, with a ε-greedy policy :
  - all nongreedy actions are given the minimal probability of selection, $\frac{\varepsilon}{|\mathcal{A}(s)|}$,
  - the greedy action is given the remaining probability, $1 - \varepsilon + \frac{\varepsilon}{|\mathcal{A}(s)|}$.

Monte Carlo Control without Exploring Starts

- ε-soft policy : $\pi(a \mid s) \geq \frac{\varepsilon}{|\mathcal{A}(s)|}, \forall s, \forall a$.
- Among ε-soft policies, ε-greedy policies are in some sense those that are closest to greedy.
- Indeed, with a ε-greedy policy :
  - all nongreedy actions are given the minimal probability of selection, $\frac{\varepsilon}{|\mathcal{A}(s)|}$,
  - the greedy action is given the remaining probability, $1 - \varepsilon + \frac{\varepsilon}{|\mathcal{A}(s)|}$.
- GPI does not require that the policy be taken all the way to a greedy policy, only that it be moved toward a greedy policy.

Monte Carlo Control without Exploring Starts

- ε-soft policy : $\pi(a \mid s) \geq \frac{\varepsilon}{|\mathcal{A}(s)|}, \forall s, \forall a$.
- Among ε-soft policies, ε-greedy policies are in some sense those that are closest to greedy.
- Indeed, with a ε-greedy policy :
  - all nongreedy actions are given the minimal probability of selection, $\frac{\varepsilon}{|\mathcal{A}(s)|}$,
  - the greedy action is given the remaining probability, $1 - \varepsilon + \frac{\varepsilon}{|\mathcal{A}(s)|}$.
- GPI does not require that the policy be taken all the way to a greedy policy, only that it be moved toward a greedy policy.
- We propose here that the policy will be moved only to an ε-greedy.

Monte Carlo Control without Exploring Starts

- ε-soft policy : $\pi(a \mid s) \geq \frac{\varepsilon}{|\mathcal{A}(s)|}, \forall s, \forall a$.
- Among ε-soft policies, ε-greedy policies are in some sense those that are closest to greedy.
- Indeed, with a ε-greedy policy :
  - all nongreedy actions are given the minimal probability of selection, $\frac{\varepsilon}{|\mathcal{A}(s)|}$,
  - the greedy action is given the remaining probability, $1 - \varepsilon + \frac{\varepsilon}{|\mathcal{A}(s)|}$.
- GPI does not require that the policy be taken all the way to a greedy policy, only that it be moved toward a greedy policy.
- We propose here that the policy will be moved only to an ε-greedy.
- We know that for any ε-soft policy, π, any ε-greedy policy with respect to q_π is guaranteed to be better than or equal to π (proof below).

Monte Carlo Control without Exploring Starts

For any ε-soft policy, π, any ε-greedy policy with respect to q_π is guaranteed to be better than or equal to π :

$$
\begin{aligned}
q_\pi(s,\pi'(s)) &= \sum_a \pi'(a|s) q_\pi(s,a) \\
&= \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) \; + \; (1-\varepsilon) \max_a q_\pi(s,a) \\
&\geq \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) \; + \; (1-\varepsilon) \sum_a \frac{\pi(a|s) - \frac{\varepsilon}{|\mathcal{A}(s)|}}{1-\varepsilon} q_\pi(s,a) \\
&= \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) \; - \; \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) \; + \; \sum_a \pi(a|s) q_\pi(s,a) \\
&= v_\pi(s).
\end{aligned}
$$

Therefore $\pi' \geq \pi$, $\quad$ *i.e.*$\forall s \in \mathcal{S}, v_{\pi'}(s) \geq v_\pi(s)$.

Monte Carlo Control without Exploring Starts

For any ε-soft policy, π, any ε-greedy policy with respect to q_π is guaranteed to be better than or equal to π :

$$
\begin{aligned}
q_\pi(s, \pi'(s)) &= \sum_a \pi'(a|s) q_\pi(s,a) \\
&= \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) \;+\; (1-\varepsilon) \max_a q_\pi(s,a) \\
&\geq \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) \;+\; (1-\varepsilon) \sum_a \frac{\pi(a|s) - \frac{\varepsilon}{|\mathcal{A}(s)|}}{1-\varepsilon} q_\pi(s,a) \\
&= \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) \;-\; \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) \;+\; \sum_a \pi(a|s) q_\pi(s,a) \\
&= v_\pi(s).
\end{aligned}
$$

Therefore $\pi' \geq \pi, \quad i.e. \forall s \in \mathcal{S}, v_{\pi'}(s) \geq v_\pi(s)$.

Indeed, we know also that equality can hold only when both π and π' are optimal among the ε-soft policies.

Monte Carlo Control without Exploring Starts

On-policy first-visit MC control (for ε-soft policies), estimates $\pi \approx \pi_*$

Algorithm parameter: small $\varepsilon > 0$

Initialize:

$\quad \pi \leftarrow$ an arbitrary ε-soft policy

$\quad Q(s,a) \in \mathbb{R}$ (arbitrarily), for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

$\quad Returns(s,a) \leftarrow$ empty list, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

Repeat forever (for each episode):

$\quad$ Generate an episode following π: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$

$\quad G \leftarrow 0$

$\quad$ Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:

$\qquad G \leftarrow \gamma G + R_{t+1}$

$\qquad$ Unless the pair S_t, A_t appears in $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:

$\qquad\quad$ Append G to $Returns(S_t, A_t)$

$\qquad\quad Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$

$\qquad\quad A^* \leftarrow \arg\max_a Q(S_t, a)$ (with ties broken arbitrarily)

$\qquad\quad$ For all $a \in \mathcal{A}(S_t)$:

$$\pi(a|S_t) \leftarrow \begin{cases} 1 - \varepsilon + \varepsilon/|\mathcal{A}(S_t)| & \text{if } a = A^* \\ \varepsilon/|\mathcal{A}(S_t)| & \text{if } a \neq A^* \end{cases}$$

Now we achieve the best policy among the ε-soft policies, but we have eliminated the assumption of exploring starts.

On-policy and off-policy learning

Off-policy learning considers two policies :

- Target policy : the one being learned and becomes the optimal policy.
- Behavior policy : which is more exploratory and used to generate behavior.

On-policy and off-policy learning

Off-policy learning considers two policies :

- Target policy : the one being learned and becomes the optimal policy.
- Behavior policy : which is more exploratory and used to generate behavior.

On-policy learning considers only one policy used to generate beahivior and is learned and becomes the optimal pilicy.

On-policy and off-policy learning

Off-policy learning considers two policies :

- Target policy : the one being learned and becomes the optimal policy.
- Behavior policy : which is more exploratory and used to generate behavior.

On-policy learning considers only one policy used to generate beahivior and is learned and becomes the optimal pilicy.

With off-policy methods, because the data is not due to the target policy, we have greater variance, and slow convergence.

Off-policy Prediction via Importance Sampling

- Suppose we like to estimate v_π or q_π for a given target policy π,

Off-policy Prediction via Importance Sampling

- Suppose we like to estimate v_π or q_π for a given target policy π,
- but we only have episodes following a given behavior policy b, $b \neq \pi$.

Off-policy Prediction via Importance Sampling

- Suppose we like to estimate v_π or q_π for a given target policy π,
- but we only have episodes following a given behavior policy b, $b \neq \pi$.
- Assuption of coverage : In order to use episodes from b to estimate values for π, we require that every action taken under π is also taken, at least occasionally, under b.

Off-policy Prediction via Importance Sampling

- Suppose we like to estimate v_π or q_π for a given target policy π,
- but we only have episodes following a given behavior policy b, $b \neq \pi$.
- Assuption of coverage : In order to use episodes from b to estimate values for π, we require that every action taken under π is also taken, at least occasionally, under b.
- Assuption of coverage : $\pi(a \mid s) > 0 \Rightarrow b(a \mid s) > 0$.

Off-policy Prediction via Importance Sampling

- Suppose we like to estimate v_π or q_π for a given target policy π,
- but we only have episodes following a given behavior policy b, $b \neq \pi$.
- Assuption of coverage : In order to use episodes from b to estimate values for π, we require that every action taken under π is also taken, at least occasionally, under b.
- Assuption of coverage : $\pi(a \mid s) > 0 \Rightarrow b(a \mid s) > 0$.
- Importance Sampling : a general technique for estimating expected values under one distribution given samples from another.

Off-policy Prediction via Importance Sampling

- Suppose we like to estimate v_π or q_π for a given target policy π,
- but we only have episodes following a given behavior policy b, $b \neq \pi$.
- Assuption of coverage : In order to use episodes from b to estimate values for π, we require that every action taken under π is also taken, at least occasionally, under b.
- Assuption of coverage : $\pi(a \mid s) > 0 \Rightarrow b(a \mid s) > 0$.
- Importance Sampling : a general technique for estimating expected values under one distribution given samples from another.
- importance-sampling ratio : We weight returns according to the relative probability of their trajectories occurring under the target and behavior policies.

Off-policy Prediction via Importance Sampling

- From a starting state S_t :

$$
\begin{aligned}
&\Pr\{A_t, S_{t+1}, A_{t+1}, \ldots, S_T \mid S_t, A_{t:T-1} \sim \pi\} \\
&\quad = \pi(A_t|S_t)p(S_{t+1}|S_t, A_t)\pi(A_{t+1}|S_{t+1}) \cdots p(S_T|S_{T-1}, A_{T-1}) \\
&\quad = \prod_{k=t}^{T-1} \pi(A_k|S_k)p(S_{k+1}|S_k, A_k),
\end{aligned}
$$

Off-policy Prediction via Importance Sampling

- From a starting state S_t :

$$\begin{aligned}
&\Pr\{A_t, S_{t+1}, A_{t+1}, \dots, S_T \mid S_t, A_{t:T-1} \sim \pi\} \\
&\quad = \pi(A_t|S_t)p(S_{t+1}|S_t, A_t)\pi(A_{t+1}|S_{t+1}) \cdots p(S_T|S_{T-1}, A_{T-1}) \\
&\quad = \prod_{k=t}^{T-1} \pi(A_k|S_k)p(S_{k+1}|S_k, A_k),
\end{aligned}$$

- The importance sampling ration is :

$$\rho_{t:T-1} \doteq \frac{\prod_{k=t}^{T-1} \pi(A_k|S_k)p(S_{k+1}|S_k, A_k)}{\prod_{k=t}^{T-1} b(A_k|S_k)p(S_{k+1}|S_k, A_k)} = \prod_{k=t}^{T-1} \frac{\pi(A_k|S_k)}{b(A_k|S_k)}.$$

which depends only on the two policies and the sequence, not on the MDP.

Off-policy Prediction via Importance Sampling

- From a starting state S_t :

$$
\begin{aligned}
&\Pr\{A_t, S_{t+1}, A_{t+1}, \ldots, S_T \mid S_t, A_{t:T-1} \sim \pi\} \\
&\quad = \pi(A_t|S_t)p(S_{t+1}|S_t, A_t)\pi(A_{t+1}|S_{t+1}) \cdots p(S_T|S_{T-1}, A_{T-1}) \\
&\quad = \prod_{k=t}^{T-1} \pi(A_k|S_k)p(S_{k+1}|S_k, A_k),
\end{aligned}
$$

- The importance sampling ration is :

$$
\rho_{t:T-1} \doteq \frac{\prod_{k=t}^{T-1} \pi(A_k|S_k)p(S_{k+1}|S_k, A_k)}{\prod_{k=t}^{T-1} b(A_k|S_k)p(S_{k+1}|S_k, A_k)} = \prod_{k=t}^{T-1} \frac{\pi(A_k|S_k)}{b(A_k|S_k)}.
$$

  which depends only on the two policies and the sequence, not on the MDP.

- The expected returns under π are :

$$
\mathbb{E}[\rho_{t:T-1} G_t \mid S_t = s] = v_\pi(s).
$$

Off-policy Prediction via Importance Sampling

- $\mathcal{T}(s)$: includes time steps that were first visits to s within their episodes.
- $T(t)$: the first time of termination following time t.
- G_t : the return after t up through $T(t)$.

Off-policy Prediction via Importance Sampling

- $\mathcal{T}(s)$: includes time steps that were first visits to s within their episodes.
- $T(t)$: the first time of termination following time t.
- G_t : the return after t up through $T(t)$.

- Ordinary importance sampling :

$$V(s) \doteq \frac{\sum_{t \in \mathcal{T}(s)} \rho_{t:T(t)-1} G_t}{|\mathcal{T}(s)|}.$$

(unbiased with unbounded variance)

Off-policy Prediction via Importance Sampling

- $\mathcal{T}(s)$: includes time steps that were first visits to s within their episodes.
- $T(t)$: the first time of termination following time t.
- G_t : the return after t up through $T(t)$.

- Ordinary importance sampling :

$$V(s) \doteq \frac{\sum_{t \in \mathcal{T}(s)} \rho_{t:T(t)-1} G_t}{|\mathcal{T}(s)|}.$$

(unbiased with unbounded variance)

- Weighted importance sampling :

$$V(s) \doteq \frac{\sum_{t \in \mathcal{T}(s)} \rho_{t:T(t)-1} G_t}{\sum_{t \in \mathcal{T}(s)} \rho_{t:T(t)-1}},$$

(biased with bounded variance)

Incremental Implementation of MC Policy Evaluation

- Given a sequance of returns $G_1, G_2, \dots, G_{n-1}$ all starting in the same state.

Incremental Implementation of MC Policy Evaluation

- Given a sequance of returns $G_1, G_2, \dots, G_{n-1}$ all starting in the same state.
- with corresponding random weight $W_i = \rho_{t_i:T(t_i)-1}$.

Incremental Implementation of MC Policy Evaluation

- Given a sequance of returns $G_1, G_2, \ldots, G_{n-1}$ all starting in the same state.
- with corresponding random weight $W_i = \rho_{t_i:T(t_i)-1}$.
- We like to estimate iteratively :

$$V_n \doteq \frac{\sum_{k=1}^{n-1} W_k G_k}{\sum_{k=1}^{n-1} W_k}, \qquad n \geq 2,$$

Incremental Implementation of MC Policy Evaluation

- Given a sequance of returns $G_1, G_2, \ldots, G_{n-1}$ all starting in the same state.
- with corresponding random weight $W_i = \rho_{t_i:T(t_i)-1}$.
- We like to estimate iteratively :

$$V_n \doteq \frac{\sum_{k=1}^{n-1} W_k G_k}{\sum_{k=1}^{n-1} W_k}, \qquad n \geq 2,$$

- We denote by C_n the cumulative sum of weights.

Incremental Implementation of MC Policy Evaluation

- Given a sequance of returns $G_1, G_2, \ldots, G_{n-1}$ all starting in the same state.
- with corresponding random weight $W_i = \rho_{t_i:T(t_i)-1}$.
- We like to estimate iteratively :

$$V_n \doteq \frac{\sum_{k=1}^{n-1} W_k G_k}{\sum_{k=1}^{n-1} W_k}, \qquad n \geq 2,$$

- We denote by C_n the cumulative sum of weights.
- and assume that V_1 is arbitrary and given.

Incremental Implementation of MC Policy Evaluation

- Given a sequance of returns $G_1, G_2, \ldots, G_{n-1}$ all starting in the same state.
- with corresponding random weight $W_i = \rho_{t_i:T(t_i)-1}$.
- We like to estimate iteratively :

$$V_n \doteq \frac{\sum_{k=1}^{n-1} W_k G_k}{\sum_{k=1}^{n-1} W_k}, \qquad n \geq 2,$$

- We denote by C_n the cumulative sum of weights.
- and assume that V_1 is arbitrary and given.
- Then the update rule is :

$$V_{n+1} \doteq V_n + \frac{W_n}{C_n}\Big[G_n - V_n\Big], \qquad n \geq 1,$$

Incremental Implementation of MC Policy Evaluation

- Given a sequance of returns $G_1, G_2, \dots, G_{n-1}$ all starting in the same state.
- with corresponding random weight $W_i = \rho_{t_i:T(t_i)-1}$.
- We like to estimate iteratively :

$$V_n \doteq \frac{\sum_{k=1}^{n-1} W_k G_k}{\sum_{k=1}^{n-1} W_k}, \qquad n \geq 2,$$

- We denote by C_n the cumulative sum of weights.
- and assume that V_1 is arbitrary and given.
- Then the update rule is :

$$V_{n+1} \doteq V_n + \frac{W_n}{C_n}\Big[G_n - V_n\Big], \qquad n \geq 1,$$

Off-policy MC Policy Evaluation

Off-policy MC prediction (policy evaluation) for estimating $Q \approx q_\pi$

Input: an arbitrary target policy π
Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
    $Q(s,a) \in \mathbb{R}$ (arbitrarily)
    $C(s,a) \leftarrow 0$

Loop forever (for each episode):
    $b \leftarrow$ any policy with coverage of π
    Generate an episode following b: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
    $G \leftarrow 0$
    $W \leftarrow 1$
    Loop for each step of episode, $t = T-1, T-2, \ldots, 0$, while $W \neq 0$:
        $G \leftarrow \gamma G + R_{t+1}$
        $C(S_t, A_t) \leftarrow C(S_t, A_t) + W$
        $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t,A_t)} [G - Q(S_t, A_t)]$
        $W \leftarrow W \frac{\pi(A_t|S_t)}{b(A_t|S_t)}$

Off-policy Monte Carlo Control

- Advanatage : the target policy may be deterministic (e.g., greedy), while the behavior policy can continue to sample all possible actions.

Off-policy Monte Carlo Control

- Advanatage : the target policy may be deterministic (e.g., greedy), while the behavior policy can continue to sample all possible actions.
- In off policy methods, we follow the behavior policy while learning about and improving the target policy.

Off-policy Monte Carlo Control

- Advanatage : the target policy may be deterministic (e.g., greedy), while the behavior policy can continue to sample all possible actions.
- In off policy methods, we follow the behavior policy while learning about and improving the target policy.
- It is required that the behavior policy has a nonzero probability of selecting all actions that might be selected by the target policy (assumption of coverage).

Off-policy Monte Carlo Control

- Advanatage : the target policy may be deterministic (e.g., greedy), while the behavior policy can continue to sample all possible actions.
- In off policy methods, we follow the behavior policy while learning about and improving the target policy.
- It is required that the behavior policy has a nonzero probability of selecting all actions that might be selected by the target policy (assumption of coverage).
- To explore all possibilities, we require that the behavior policy be soft (i.e., that it select all actions in all states with nonzero probability).

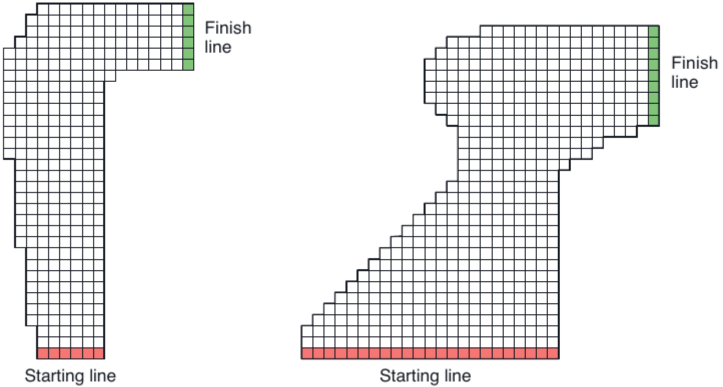
Off-policy Monte Carlo Control

Off-policy MC control, for estimating $\pi \approx \pi_*$

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
- $Q(s,a) \in \mathbb{R}$ (arbitrarily)
- $C(s,a) \leftarrow 0$
- $\pi(s) \leftarrow \arg\max_a Q(s,a)$ (with ties broken consistently)

Loop forever (for each episode):
- $b \leftarrow$ any soft policy
- Generate an episode using b: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
- $G \leftarrow 0$
- $W \leftarrow 1$
- Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
  - $G \leftarrow \gamma G + R_{t+1}$
  - $C(S_t, A_t) \leftarrow C(S_t, A_t) + W$
  - $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)} [G - Q(S_t, A_t)]$
  - $\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$ (with ties broken consistently)
  - If $A_t \neq \pi(S_t)$ then exit inner Loop (proceed to next episode)
  - $W \leftarrow W \frac{1}{b(A_t|S_t)}$

Racetrack exercise



Racetrack exercise

Exercise 5.12: Racetrack (programming) Consider driving a race car around a turn like those shown in Figure 5.5. You want to go as fast as possible, but not so fast as to run off the track. In our simplified racetrack, the car is at one of a discrete set of grid positions, the cells in the diagram. The velocity is also discrete, a number of grid cells moved horizontally and vertically per time step. The actions are increments to the velocity components. Each may be changed by $+1$, -1, or 0 in each step, for a total of nine (3×3) actions. Both velocity components are restricted to be nonnegative and less than 5, and they cannot both be zero except at the starting line. Each episode begins in one of the randomly selected start states with both velocity components zero and ends when the car crosses the finish line. The rewards are -1 for each step until the car crosses the finish line. If the car hits the track boundary, it is moved back to a random position on the starting line, both velocity components are reduced to zero, and the episode continues. Before updating the car's location at each time step, check to see if the projected path of the car intersects the track boundary. If it intersects the finish line, the episode ends; if it intersects anywhere else, the car is considered to have hit the track boundary and is sent back to the starting line. To make the task more challenging, with probability 0.1 at each time step the velocity increments are both zero, independently of the intended increments. Apply a Monte Carlo control method to this task to compute the optimal policy from each starting state. Exhibit several trajectories following the optimal policy (but turn the noise off for these trajectories). □

Thank you !