Data Science

Reinforcement Learning

Lionel Fillatre
Polytech Nice Sophia
lionel.fillatre@univ-cotedazur.fr

Outline of this Lecture

- Quick Recap
- Monte Carlo Learning
- Monte Carlo Control
- On-Policy
- Off-Policy
- Conclusion

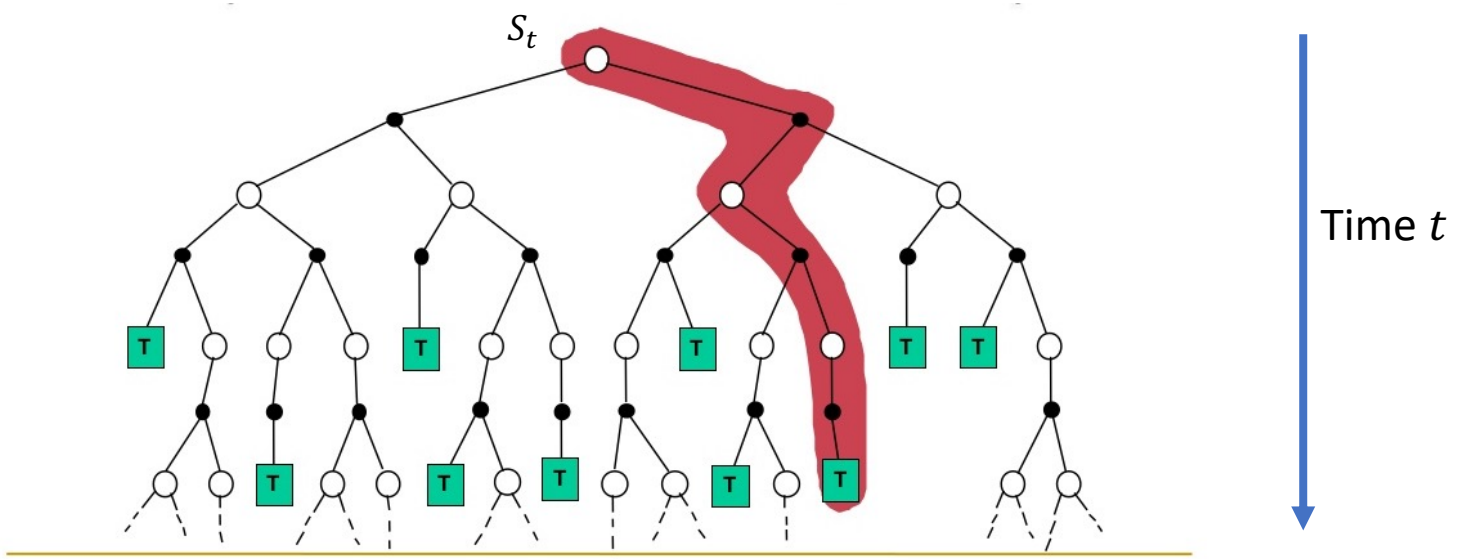
Quick Recap: When dynamics are known…

- So far, we've focused on planning, where the dynamics are known.
- The optimal value functions are characterized in terms of a Bellman fixed point update.
- Since the Bellman operator is a contraction map, we can just keep applying it repeatedly, and we'll converge to a unique fixed point.
- **What are the limitations of value/policy iteration?**
  - Assumes known dynamics
  - The Markov property holds
  - Requires explicitly representing q^* or v^* as a vector
  - The number of states $|\mathcal{S}|$ can be extremely large, or infinite
  - The number of actions $|\mathcal{A}|$ can be infinite (e.g. continuous voltages in a machine)
- But value iteration is still a foundation for a lot of more practical RL algorithms.

Monte Carlo Learning

Monte Carlo (MC) Methods

- Monte Carlo methods are learning methods
  - Experience ⇒ values, policy
- Monte Carlo methods learn from complete sampled trajectories
  - Only defined for episodic tasks
  - All episodes must terminate
- Monte Carlo uses the simplest possible idea: value = mean return



Monte-Carlo Policy Evaluation

- Remember that the value function is the expected return:

$$v_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s]$$

- Remember that the return is the total discounted reward:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^{T-1} R_T$$

- Goal: learn v_π from episodes of experience under policy π:

$$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, \dots, S_k \text{ when } A_1, A_2, A_3, \dots \sim \pi$$

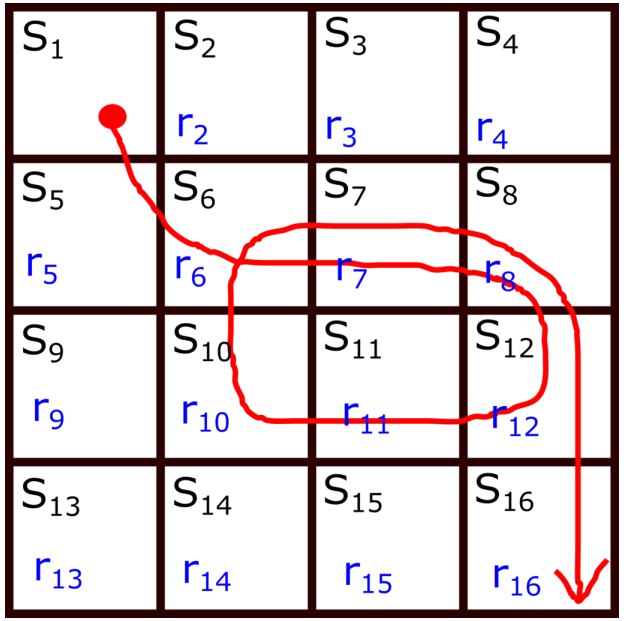
- **Monte-Carlo policy evaluation uses empirical mean instead of expectation $\mathbb{E}_{\boldsymbol{\pi}}[\cdot]$**

Monte-Carlo Policy Evaluation

- Goal: learn $v_\pi(s)$ from episodes of experience under policy π:
- Idea: Average returns observed after visits to s:

- Every-Visit MC: average returns for every time s is visited in an episode
- First-visit MC: average returns only for first time s is visited in an episode
- Both converge asymptotically based on the law of large numbers

Frozen Lake Environment



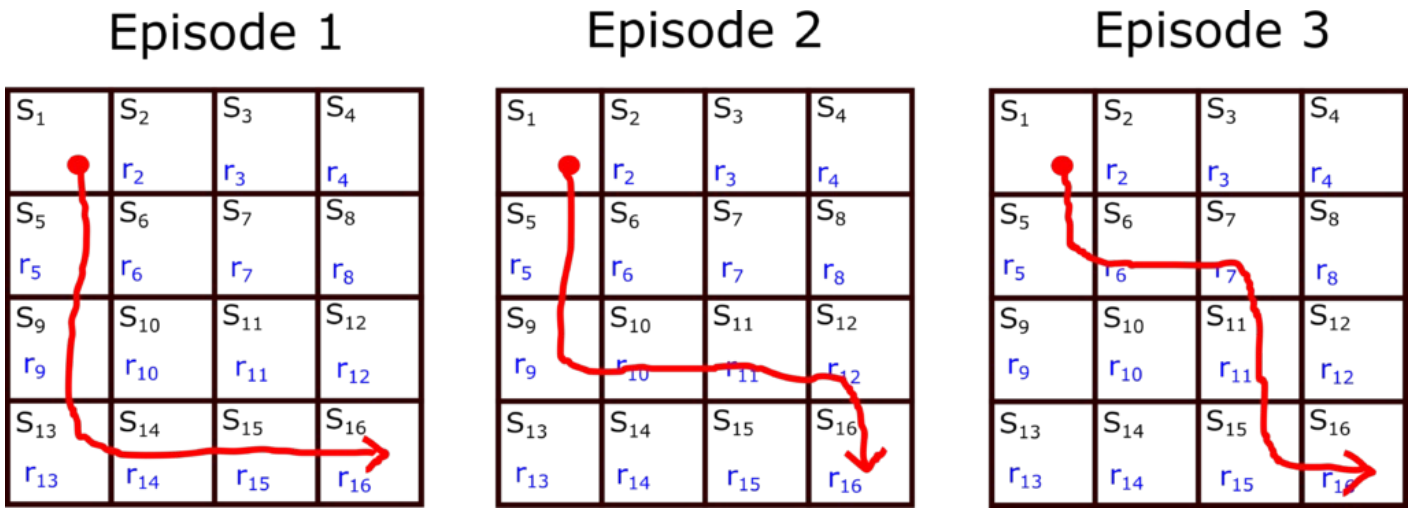
- The start state is s_1. The end state is the state s_{16}.
- The episode is described by the trajectory

$$S_0 = s_1, S_1 = s_5, S_2 = s_6, S_3 = s_7, S_4 = s_8, S_5 = s_{12}, S_6 = s_{11}, S_7 = s_{10}, S_8 = s_6, S_9 = s_7, S_{10} = s_8, S_{11} = s_{12}, S_{12} = s_{16}$$

- Each occurrence of the state s_i in an episode is called a visit. There are two visits of s_7
- The obtained rewards in the sequence are

$$R_1 = r_5, R_2 = r_6, R_3 = r_7, R_4 = r_8, R_5 = r_{12}, R_6 = r_{11}, R_7 = r_{10}, R_8 = r_6, R_9 = r_7, R_{10} = r_8, R_{11} = r_{12}, R_{12} = r_{16}$$

- Since we only have a single episode, the average return for every state will be equal to the return obtained at this single episode (first visit, $\gamma = 1$).
  - $\hat{v}_\pi(s_1) = G(s_1)^{episode\ 1} = R_1 + R_2 + R_3 + R_4 + R_5 + R_6 + R_7 + R_8 + R_9 + R_{10} + R_{11} + R_{12}$
    $= r_5 + r_6 + r_7 + r_8 + r_{12} + r_{11} + r_{10} + r_6 + r_7 + r_8 + r_{12} + r_{16}$
  - $\hat{v}_\pi(s_6) = G(s_6)^{episode\ 1} = R_3 + R_4 + R_5 + R_6 + R_7 + R_8 + R_9 + R_{10} + R_{11} + R_{12}$
    $= r_7 + r_8 + r_{12} + r_{11} + r_{10} + r_6 + r_7 + r_8 + r_{12} + r_{16}$



- The returns from the state s_1 for these three episodes is

$$G(s_1)^{episode\ 1} = r_5 + r_9 + r_{13} + r_{14} + r_{15} + r_{16}$$
$$G(s_1)^{episode\ 2} = r_5 + r_9 + r_{10} + r_{11} + r_{12} + r_{16}$$
$$G(s_1)^{episode\ 3} = r_5 + r_6 + r_7 + r_{11} + r_{15} + r_{16}$$

- Since the state s_1 is visited 3 times, we have

$$\hat{v}_\pi(s_1) = \frac{G(s_1)^{episode\ 1} + G(s_1)^{episode\ 2} + G(s_1)^{episode\ 3}}{3}$$

- The returns from the state s_9 for these three episodes is

$$G(s_9)^{episode\ 1} = r_{13} + r_{14} + r_{15} + r_{16}$$
$$G(s_9)^{episode\ 2} = r_{10} + r_{11} + r_{12} + r_{16}$$

No return for episode 3

- Since the state s_9 is visited 2 times, we have

$$\hat{v}_\pi(s_9) = \frac{G(s_9)^{episode\ 1} + G(s_9)^{episode\ 2}}{2}$$

https://aleksandarhaber.com/monte-carlo-method-for-learning-state-value-functions-first-visit-method-reinforcement-learning-tutorial/

First-Visit Monte Carlo Evaluation

- Input: a policy π to be evaluated

- Initialize:
  - $V(s) \in \mathbb{R}$ arbitrarily for all $s \in \mathcal{S}$
  - $Returns(s) \leftarrow$ empty list, for all $s \in \mathcal{S}$

- Loop forever (for each episode):
  - Generate an episode following π: $S_0, A_0, R_1, S_1, A_1, R_2, \dots, S_{T-1}, A_{T-1}, R_T$
  - $G \leftarrow 0$
  - Loop for each step of episode, $t = T-1, T-2, \dots, 0$:
    - $G \leftarrow \gamma G + R_{t+1}$
      If S_t does not appear in $S_0, S_1, \dots, S_{t-1}$ (first visit of S_t):
      - Append G to $Returns(S_t)$
      - $V(S_t) \leftarrow average(Returns(S_t))$

First-Visit Monte Carlo Evaluation

- To evaluate state s from several episodes

- The first time-step t that state s is visited in an episode
  - Incrementer counter for each episode: $N(s) \leftarrow N(s) + 1$
  - Increment total return: $G(s) \leftarrow G(s) + G_t$

- Value is estimated by mean return: $\hat{v}(s) = \frac{G(s)}{N(s)}$

- By law of large numbers: $\hat{v}(s) \rightarrow v_\pi(s)$ as $N(s) \rightarrow \infty$

Incremental Mean

- The mean μ_k of a sequence $x_1, \ldots, x_k$ can be computed incrementally:

$$\begin{aligned}\mu_k &= \frac{1}{k}\sum_{j=1}^{k} x_j \\ &= \frac{1}{k}\left(x_k + \sum_{j=1}^{k-1} x_j\right) \\ &= \frac{1}{k}\left(x_k + (k-1)\mu_{k-1}\right) \\ &= \mu_{k-1} + \frac{1}{k}\left(x_k - \mu_{k-1}\right)\end{aligned}$$

Incremental Monte Carlo Evaluation

- Update $\hat{v}(s)$ incrementally after episode $S_0, A_0, R_1, S_1, A_1, R_2, \dots, S_k$ and visits

- For each state S_t with return G_t
  - $N(S_t) \leftarrow N(S_t) + 1$
  - $\hat{v}(S_t) = \frac{G(S_t)}{N(S_t)} \leftarrow \hat{v}(S_t) + \frac{1}{N(S_t)}\left(G_t - \hat{v}(S_t)\right)$

- In non-stationary problems, it can be useful to track a running mean, i.e. forget old episodes

$$\hat{v}(S_t) \leftarrow \hat{v}(S_t) + \alpha\left(G_t - \hat{v}(S_t)\right)$$

Every-Visit Monte Carlo Evaluation

- To evaluate state s from several episodes and from several visits

- Every time-step t that state s is visited in an episode
  - Incrementer counter for each episode: $N(s) \leftarrow N(s) + 1$
  - Increment total return: $G(s) \leftarrow G(s) + G_t$

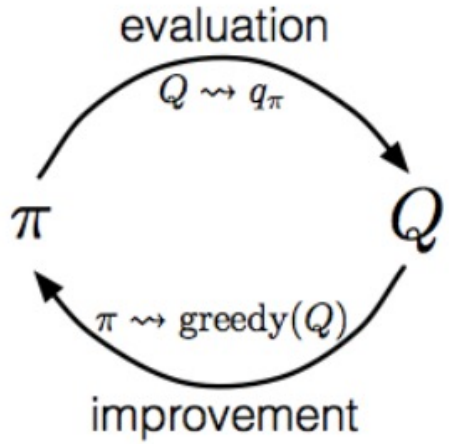
- Value is estimated by mean return: $\hat{v}(s) = \frac{G(s)}{N(s)}$

- By law of large numbers: $\hat{v}(s) \rightarrow v_\pi(s)$ as $N(s) \rightarrow \infty$

Monte Carlo Control

Monte-Carlo Control

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



Greedy policy: $\pi(s) = \underset{a' \in A}{\text{argmax}}\, Q(s, a')$

- **MC policy iteration step:** Policy evaluation using MC methods followed by policy improvement
- **Policy improvement step:** greedify with respect to value (or state-action value) function

MC Estimation of Action Values

- MC is most useful when a model is not available
  - We want to learn $q^*(s,a)$ because we can get an optimal policy without knowing dynamics
  - **Reminder:** $q_\pi(s,a)$ - average return starting from state s and action a following π

$$q_\pi(s,a) = r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a) v_\pi(s')$$

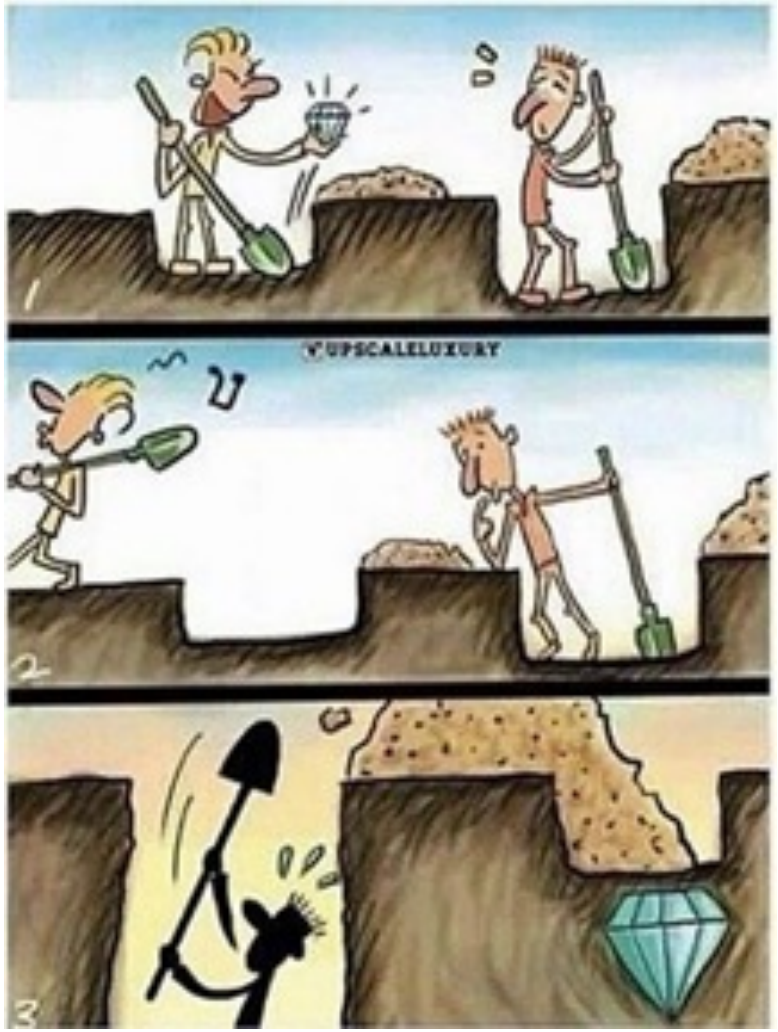
- Converges asymptotically if every state-action pair is visited.
- Is this possible if we are using a deterministic policy?

The Exploration Problem

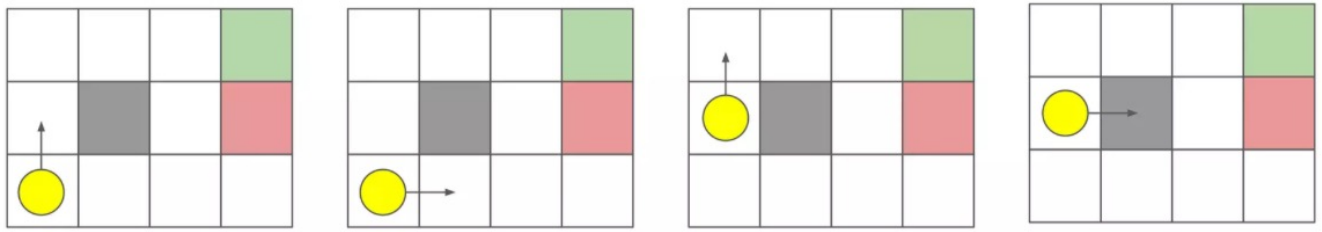
- If we always follow the deterministic policy to collect experience, we will never have the opportunity to see and evaluate (estimate q) of alternative actions.

- All learning methods face a dilemma: they seek to learn action values conditioned on subsequent optimal behaviour but they need to act suboptimally in order to explore all actions (to discover the optimal actions). **This is the exploration-exploitation dilemma.**

- Solutions (discussed later):
  - **Exploring starts**: Every state-action pair has a non-zero probability of being the starting pair
  - Give up on deterministic policies and only search over **ϵ-soft policies**
  - **Off-policy**: use a different policy to collect experience than the one you care to evaluate



Monte Carlo Exploring Starts



Several (S_0, A_0)

- 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, $t = T-1, T-2, \ldots, 0$:
    - $G \leftarrow \gamma G + R_{t+1}$
      If the the pair S_t, A_t does appear in $S_0, A_0, R_1, \ldots, S_{t-1}, A_{t-1}$:
      - Append G to $Returns(S_t, A_t)$
      - $Q(S_t, A_t) \leftarrow average(Returns(S_t, A_t))$
      - $\pi(S_t) \leftarrow \underset{u \in A}{\operatorname{argmax}}\, Q(S_t, u)$

Convergence of MC Control

- Greedified policy meets the conditions for policy improvement:

$$\begin{aligned} q_{\pi_k}\big(s, \pi_{k+1}(s)\big) &= q_{\pi_k}\left(s, \operatorname*{argmax}_{a' \in \mathcal{A}} q_{\pi_k}(s, a')\right) \\ &= \max_{a' \in \mathcal{A}} q_{\pi_k}(s, a') \\ &\geq q_{\pi_k}\big(s, \pi_k(s)\big) \\ &= v_{\pi_k}(s) \end{aligned}$$

- And thus π_{k+1} must be better that π_k or as good as π_k.

- This assumes **exploring starts** and **infinite number of episodes** for MC policy evaluation

Monte Carlo Control without Exploring Starts

- How can we avoid the unlikely assumption of exploring starts?
- The only general way to ensure that all actions are selected infinitely often is for the agent to continue to select them.
- There are two approaches to ensuring this, resulting in what we call **on-policy methods** and **off-policy methods**.
  - On-policy methods attempt to evaluate or improve the policy that is used to make decisions,
  - Off-policy methods evaluate or improve a policy different from that used to generate the data.

On-Policy

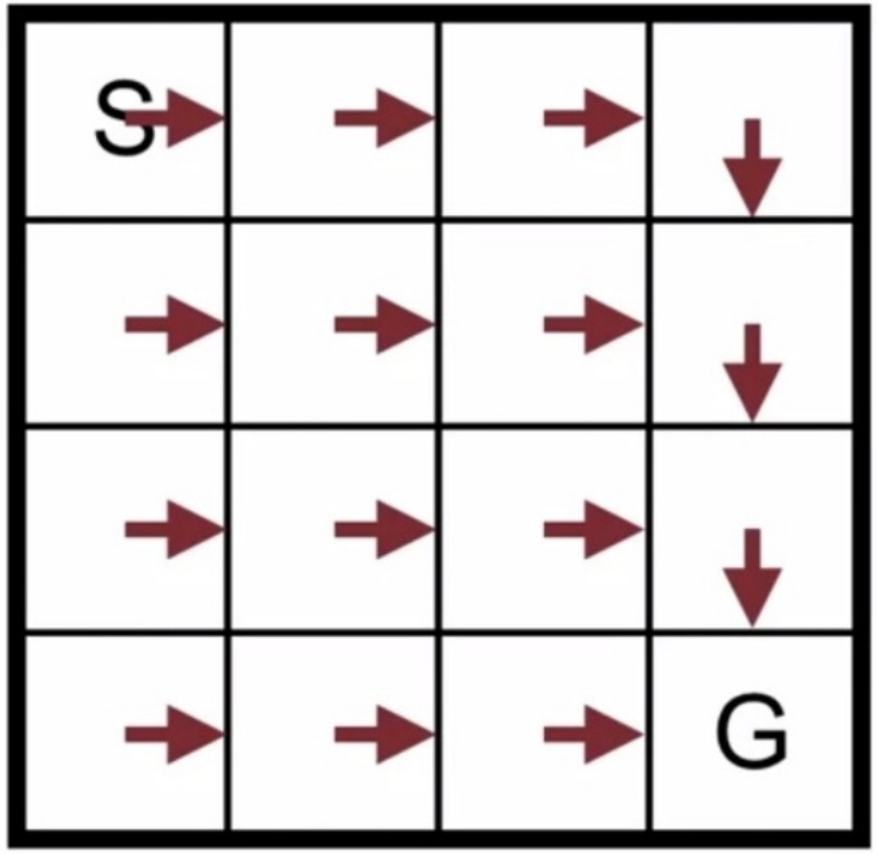
On-policy Monte Carlo Control

- On-policy: learn about policy currently executing
- The policy must be eternally soft: $\pi(a|s) > 0$ for all s and $a \in \mathcal{A}(s)$ where $\mathcal{A}(s)$ is the set of actions possible from state s.
- An ϵ-soft policy is defined as a soft policy such as $\pi(a|s) \geq \frac{\epsilon}{|\mathcal{A}(s)|}$ for all states and actions
- For example, let us consider the ϵ-greedy policy. Its probability $\pi(a|s)$ is defined by

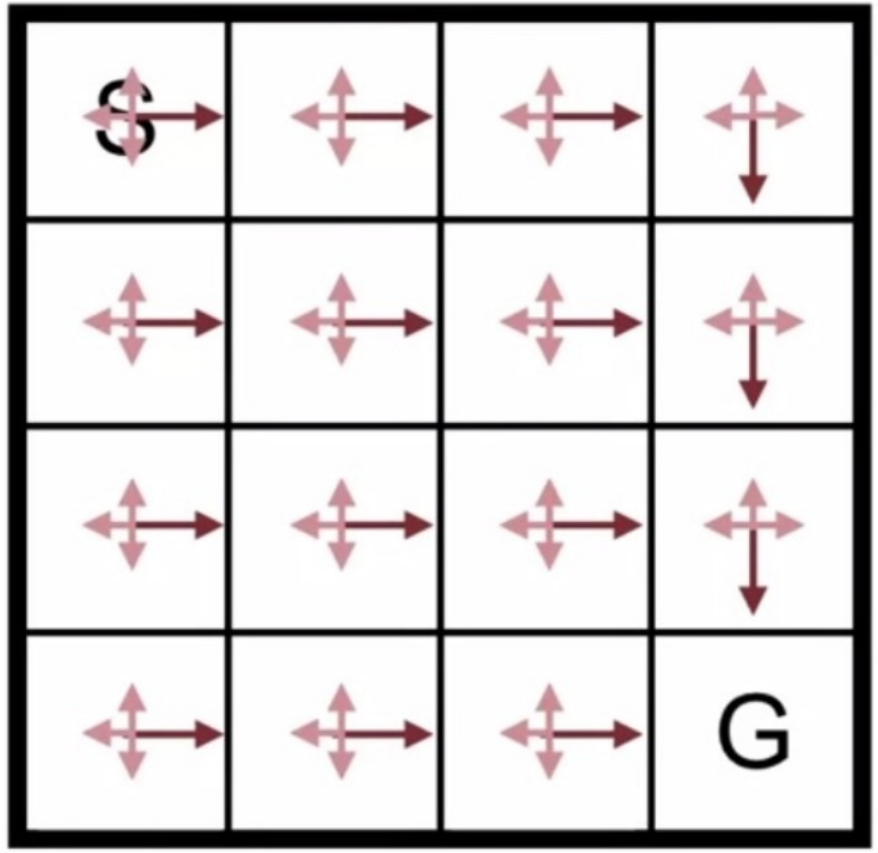
$$\pi(a|s) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{|\mathcal{A}(s)|} & \text{if } a \in \mathcal{A}(s) \text{ is the greedy action} \\ \frac{\epsilon}{|\mathcal{A}(s)|} & \text{if } a \in \mathcal{A}(s) \text{ is a non} - \text{greedy action} \end{cases}$$

- On-policy first-visit MC control algorithm (described on the next slide) converges to the best ϵ-soft policy.

Illustration of ϵ-greedy policy



Optimal policy π^*



ϵ-greedy policy

On-policy first-visit MC control (for ϵ-soft policies)

- Initialize:
  - A small parameter $\epsilon > 0$
  - π an arbitraty ϵ-soft policy such that $\pi(a|s) \geq \frac{\epsilon}{|\mathcal{A}(s)|}$ for all states and actions
  - $Q(s,a) \in \mathbb{R}$ 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):
  - Generate an episode 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, $t = T-1, T-2, \ldots, 0$:
    - $G \leftarrow \gamma G + R_{t+1}$
    - If the pair S_t, A_t does not appear in $S_0, A_0, R_1, \ldots, S_{t-1}, A_{t-1}$:
      - Append G to $Returns(S_t, A_t)$
      - $Q(S_t, A_t) \leftarrow average(Returns(S_t, A_t))$
      - $A^* \leftarrow \underset{a' \in A}{\text{argmax}}\, Q(S_t, a')$ (with ties broken arbitrarily)
      - For all $a \in \mathcal{A}(S_t)$

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

Improvement of ϵ-greedy policy

- 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.
- Let π' be the ϵ -greedy policy

$$q_\pi\big(s,\pi'(s)\big) = \sum_{a\in A}\pi'(a|s)q_\pi(s,a)$$

$$= \frac{\epsilon}{|\mathcal{A}(s)|}\sum_{a\in A} q_\pi(s,a) + (1-\epsilon)\max_{a'\in A} q_\pi(s,a')$$

$$\geq \frac{\epsilon}{|\mathcal{A}(s)|}\sum_{a\in A} q_\pi(s,a) + (1-\epsilon)\sum_{a\in A}\frac{\pi(a|s) - \frac{\epsilon}{|\mathcal{A}(s)|}}{1-\epsilon} q_\pi(s,a)$$

$$= \frac{\epsilon}{|\mathcal{A}(s)|}\sum_{a\in A} q_\pi(s,a) - \frac{\epsilon}{|\mathcal{A}(s)|}\sum_{a\in A} q_\pi(s,a) + \sum_{a\in A}\pi(a|s)q_\pi(s,a)$$

$$= v_\pi(s)$$

since $\sum_{a\in A}\pi(a|s) - \frac{\epsilon}{|\mathcal{A}(s)|} = 1-\epsilon$, $\pi(a|s) - \frac{\epsilon}{|\mathcal{A}(s)|} \geq 0$ since π is an ϵ-soft policy, and $\max_{a'\in A} q_\pi(s,a')$ larger than any convex combination of $q_\pi(s,a)$

Off-Policy

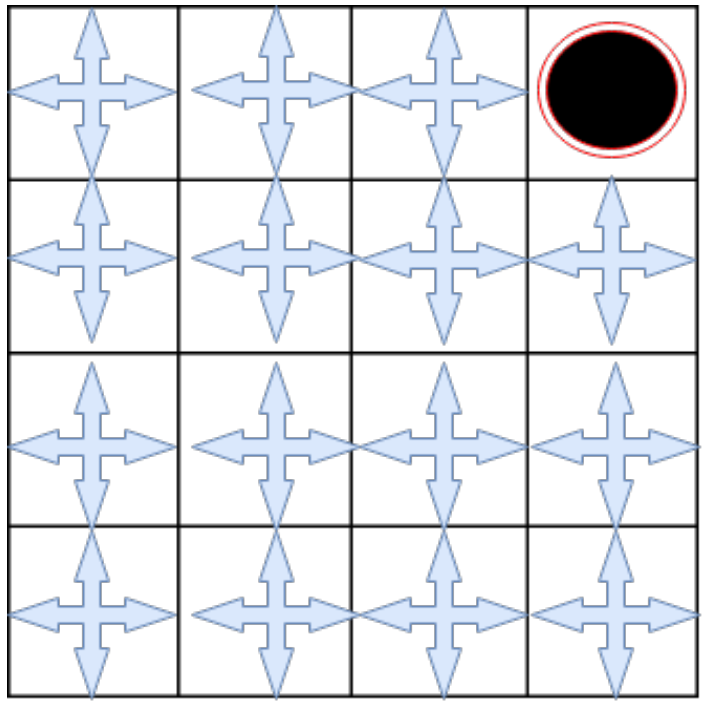
Off-Policy Methods

- Learn the value of the target policy π from experience due to behavior policy μ
- For example, π is the greedy policy (and ultimately the optimal policy) while μ is exploratory (e.g., ϵ-soft) policy
- In general, we only require coverage, i.e., that μ generates behavior that covers or include π

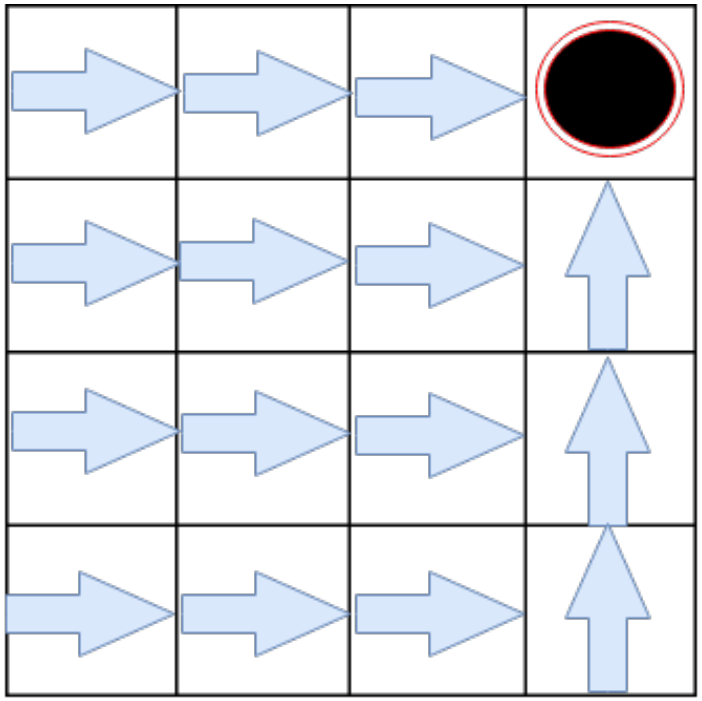
$$\mu(a|s) > 0 \text{ for every } s, a \text{ at which } \pi(a|s) > 0$$

- Can we average returns as before to obtain the value function of π ?
- Idea: Importance Sampling
  - Weight each return by the ratio of the probabilities of the trajectory under the two policies.

Off-Policy Illustration



Behavior Policy



Target Policy

Importance Sampling: Principle

- Estimate the expectation of a distribution $\pi(X)$ from an other proposal distribution $\mu(X)$

$$\mathbb{E}_{X\sim\pi}[f(X)] = \int f(X)\pi(X)dX = \int f(X)\frac{\pi(X)}{\mu(X)}\mu(X)dX = \mathbb{E}_{X\sim\mu}\left[\frac{\pi(X)}{\mu(X)}f(X)\right]$$

- The two expectations are equivalent provided we use an appropriate weight
- Generally, we choose an easy-to-sample proposal distribution $\mu(X)$

- In practice, we get

$$\mathbb{E}_{X\sim\pi}[f(X)] \approx \frac{1}{N}\sum_{i=1}^{N}\frac{\pi\left(x^{(i)}\right)}{\mu\left(x^{(i)}\right)}f\left(x^{(i)}\right) = \frac{1}{N}\sum_{i=1}^{N}w^{(i)}f\left(x^{(i)}\right)$$

when we have N i.i.d. samples $x^{(i)}\sim\mu(X)$ of X generated from $\mu(X)$

- The quantities $w^{(i)} = \frac{\pi\left(x^{(i)}\right)}{\mu\left(x^{(i)}\right)}$ are known as importance weights

Importance Sampling Ratio for Off-Policy

- Probability of the trajectory, after state S_t until the terminal state S_T, under policy π:

$$\mathbb{P}(A_t, S_{t+1}, A_{t+1}, \cdots, S_T | S_t, A_t \sim \pi, \cdots, A_{T-1} \sim \pi) = \pi(A_t|S_t)p(S_{t+1}|S_t, A_t)\pi(A_{t+1}|S_{t+1}) \cdots p(S_T|S_{T-1}, A_{T-1})$$

$$= \prod_{k=t}^{T-1} \pi(A_k|S_k)p(S_{k+1}|S_k, A_k)$$

- **Importance sampling:** each return is weighted by the relative probability of the trajectory under the target and behavior policies

$$\rho_t^{T-1} = \frac{\prod_{k=t}^{T-1} \pi(A_k|S_k)p(S_{k+1}|S_k, A_k)}{\prod_{k=t}^{T-1} \mu(A_k|S_k)p(S_{k+1}|S_k, A_k)} = \prod_{k=t}^{T-1} \frac{\pi(A_k|S_k)p(S_{k+1}|S_k, A_k)}{\mu(A_k|S_k)p(S_{k+1}|S_k, A_k)} = \prod_{k=t}^{T-1} \frac{\pi(A_k|S_k)}{\mu(A_k|S_k)}$$

Importance Sample for Off-Policy

- Notations
  - To simplify notations, it is convenient to assume that all the episodes are concatenated in a single sequence
    - If the first episode ends in a terminal state at time $t = 100$, than the next episode begins at time $t = 101$
  - Let $\mathcal{T}(s)$ the set of time steps that were first visits to s within their episodes
  - Let $T(t)$ denote the first time of termination following t
  - Let G_t denote the return after t up through $T(t)$ when we use the behavior policy μ to choose the actions

- Remember that the **value function** (it is simpler to present that the state-action value function) is the expected return:

$$v_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s] = \mathbb{E}_\mu[\rho_t^{T-1} G_t|S_t = s]$$

- Then
  - $\{G_t\}_{t\in\mathcal{T}(s)}$ are the returns that pertain to state s
  - $\left\{\rho_t^{T(t)-1}\right\}_{t\in\mathcal{T}(s)}$ are the corresponding importance-sampling ratios

- Ordinary importance sampling estimate:

$$\hat{v}(s) = \frac{\sum_{t\in\mathcal{T}(s)} \rho_t^{T(t)-1} G_t}{|\mathcal{T}(s)|}$$

Off-Policy Monte-Carlo in Practice

- Use returns generated from μ to evaluate π
- Multiply importance sampling corrections along whole episode

$$G_t^{\pi/\mu} = \frac{\pi(A_t|S_t)\pi(A_{t+1}|S_{t+1})\cdots\pi(A_{T-1}|S_{T-1})}{\mu(A_t|S_t)\mu(A_{t+1}|S_{t+1})\cdots\mu(A_{T-1}|S_{T-1})} G_t = \rho_t^{T-1} G_t$$

- Update value towards corrected return

$$\hat{v}(S_t) \leftarrow \hat{v}(S_t) + \alpha\left(G_t^{\pi/\mu} - \hat{v}(S_t)\right)$$

- Can not be used if μ is zero when π is non-zero
- Importance sampling can dramatially increase variance

Self-normalized importance sampling

- Often, the density we want to sample from is only known up to an unknown constant, i.e., $\mu(x) = \frac{\tilde{\mu}(x)}{\mathcal{Z}_\mu}$ where $\tilde{\mu}(x)$ is known, but $\mathcal{Z}_\mu$ is not. Of course $\mathcal{Z}_\mu$ is determined by the requirement that the integral of $\mu(x)$ be 1, but we may not be able to compute the integral $\int \tilde{\mu}(X)dX = \mathcal{Z}_\mu$.

- Let our proposal be of the form $\mu(x) = \frac{\tilde{\mu}(x)}{\mathcal{Z}_\mu}$

$$\mathbb{E}_{X\sim\pi}[f(X)] = \int f(X)\pi(X)dX = \int f(X)\frac{\pi(X)}{\mu(X)}\mu(X)dX = \frac{\mathcal{Z}_\mu}{\mathcal{Z}_\pi}\int f(X)\frac{\tilde{\pi}(X)}{\tilde{\mu}(X)}\mu(X)dX \approx \frac{1}{N}\frac{\mathcal{Z}_\mu}{\mathcal{Z}_\pi}\sum_{i=1}^{N}\frac{\tilde{\pi}\left(x^{(i)}\right)}{\tilde{\mu}(x^{(i)})}f\left(x^{(i)}\right) = \frac{1}{N}\frac{\mathcal{Z}_\mu}{\mathcal{Z}_\pi}\sum_{i=1}^{N} w^{(i)}f\left(x^{(i)}\right)$$

where $\int \tilde{\pi}(X)dX = \mathcal{Z}_\pi$ such that $\tilde{\pi}(X)$ may also not normalized

- How to deal with the normalization constants? We note that

$$\frac{\mathcal{Z}_\pi}{\mathcal{Z}_\mu} = \frac{1}{\mathcal{Z}_\mu}\int \tilde{\pi}(X)dX = \frac{1}{\mathcal{Z}_\mu}\int \frac{\tilde{\pi}(X)}{\mu(X)}\mu(X)dX = \int \frac{\tilde{\pi}(X)}{\tilde{\mu}(X)}\mu(X)dX \approx \frac{1}{N}\sum_{i=1}^{N}\frac{\tilde{\pi}\left(x^{(i)}\right)}{\tilde{\mu}(x^{(i)})} = \frac{1}{N}\sum_{i=1}^{N} w^{(i)}$$

- It follows that

$$\mathbb{E}_{X\sim\pi}[f(X)] \approx \sum_{i=1}^{N}\frac{w^{(i)}}{\sum_{i=1}^{N} w^{(i)}}f\left(x^{(i)}\right), \qquad w^{(i)} = \frac{\tilde{\pi}\left(x^{(i)}\right)}{\tilde{\mu}(x^{(i)})}, \qquad x^{(i)}\sim\mu(X)$$

Incremental implementation

- Let $G_i(s)$ denote the return after t_i up through $T(t_i)$ when we use the behavior policy μ to choose the actions starting from state s
- Let $G_1(s), G_2(s), \ldots, G_N(s)$ be the sequence of returns

- Remember that:

$$v_\pi(s) = \mathbb{E}_\mu[\rho_t^{T-1} G_t | S_t = s] \approx \hat{v}_{N+1}(s) = \sum_{i=1}^{N} \frac{w^{(i)}}{\sum_{i=1}^{N} w^{(i)}} G_i(s)$$

$$w^{(i)} = \rho_{t_i}^{T(t_i)-1}$$

- Incremental formula (used in practice for large N):

$$\hat{v}_{n+1}(s) = \hat{v}_n(s) + \frac{w^{(n)}}{c^{(n)}}\left(G_n(s) - \hat{v}_n(s)\right), \qquad n \geq 1$$

$$c^{(n+1)} = c^{(n)} + w^{(n+1)}$$

$$c^{(0)} = 0$$

$$\hat{v}_0(s) \in \mathbb{R}$$

Which target function to estimate?

- The off-policy approach works both for estimating the value function and the state-action value function

- In practice the self-normalized importance sampling is prefered
  - It is biased
  - But its variance is smaller

- The next slide shows the full algorithm for estimating the state-action value function

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

- Input: π an arbitrary target policy

- Initialize:
  - $Q(s,a) \in \mathbb{R}$ arbitrarily for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$
  - $C(s,a) \leftarrow 0$ for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

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

Conclusion

Monte Carlo Advantages

- MC has several advantages over dynamic programming:
  - Can learn directly from interaction with environment
  - No need for full models

- MC methods provide an alternate policy evaluation process
- One issue to watch for: maintaining sufficient exploration
- Looked at distinction between on-policy and off-policy methods