Monte Carlo Methods

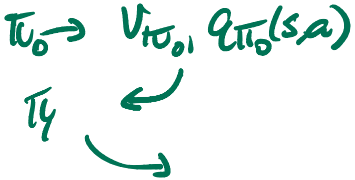
Madhavan Mukund

https://www.cmi.ac.in/~madhavan

Advanced Machine Learning
September–December 2021

Monte Carlo methods

- If we know the model, use generalized policy iteration (dynamic programming) to approximate π_*, v_*
- What if the model is a black box?



Monte Carlo methods

- If we know the model, use generalized policy iteration (dynamic programming) to approximate π_*, v_*
- What if the model is a black box?
- Generate random episodes to estimate the given quantities
- Learning through experience

$\pi \rightarrow v_\pi, q_\pi(s,a)$

↑

sampling

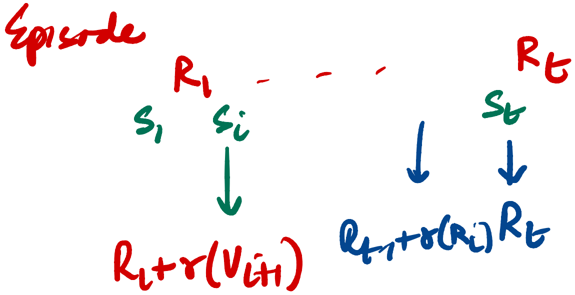
Monte Carlo methods

- If we know the model, use generalized policy iteration (dynamic programming) to approximate π_*, v_*
- What if the model is a black box?
- Generate random episodes to estimate the given quantities
- Learning through experience
- Monte Carlo algorithms — compute estimates through random sampling

Monte Carlo policy evaluation — estimating v_π

- Estimate v_π for a given policy π

- Estimate v_π for a given policy π
- Generate an episode following π, compute $v_\pi(s)$ backwards from end



Monte Carlo policy evaluation — estimating v_π

- Estimate v_π for a given policy π
- Generate an episode following π, compute $v_\pi(s)$ backwards from end
- Average out values across episodes

Monte Carlo policy evaluation — estimating v_π

- Estimate v_π for a given policy π
- Generate an episode following π, compute $v_\pi(s)$ backwards from end
- Average out values across episodes

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 — compute average for first visit to s in each episode

Monte Carlo policy evaluation — estimating v_π

- Estimate v_π for a given policy π
- Generate an episode following π, compute $v_\pi(s)$ backwards from end
- Average out values across episodes

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: $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$

  $G \leftarrow 0$

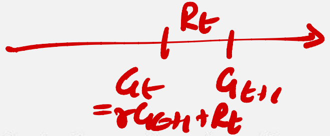
  Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:

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

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

      Append G to $Returns(S_t)$

      $V(S_t) \leftarrow$ average$(Returns(S_t))$



First visit

- First-visit MC — compute average for first visit to s in each episode
- Every-visit MC — remove Unless condition

Action value estimate — $q_\pi(s, a)$

- In the absence of a model, useful to directly estimate action values $q_\pi(s, a)$
  - Policy improvement requires $q_\pi(s, a)$
  - $v_\pi(s) = \sum_a \pi(a \mid s) q_\pi(s, a)$

Action value estimate — $q_\pi(s, a)$

- In the absence of a model, useful to directly estimate action values $q_\pi(s, a)$
  - Policy improvement requires $q_\pi(s, a)$
  - $v_\pi(s) = \sum_a \pi(a \mid s) q_\pi(s, a)$
- Need to ensure that all pairs (s, a) are visited

Action value estimate — $q_\pi(s,a)$

- In the absence of a model, useful to directly estimate action values $q_\pi(s,a)$
  - Policy improvement requires $q_\pi(s,a)$
  - $v_\pi(s) = \sum_a \pi(a \mid s) q_\pi(s,a)$
- Need to ensure that all pairs (s,a) are visited
- Exploring starts Each pair (s,a) has non-zero probability of being start of an episode
  - Will look at ways to avoid this

Action value estimate — $q_\pi(s, a)$

- In the absence of a model, useful to directly estimate action values $q_\pi(s, a)$
  - Policy improvement requires $q_\pi(s, a)$
  - $v_\pi(s) = \sum_a \pi(a \mid s) q_\pi(s, a)$
- Need to ensure that all pairs (s, a) are visited
- Exploring starts Each pair (s, a) has non-zero probability of being start of an episode
  - Will look at ways to avoid this
- With exploring starts, algorithm for estimating q_π is similar to the one for v_π

Monte Carlo Policy Iteration

- As before, alternate between policy evaluation and policy improvement

$$\pi_0 \xrightarrow{\text{evaluate}} q_{\pi_0} \xrightarrow{\text{improve}} \pi_1 \xrightarrow{\text{evaluate}} q_{\pi_1} \xrightarrow{\text{improve}} \pi_2 \xrightarrow{\text{evaluate}} \cdots$$

q_π ≠ v_π

Monte Carlo Policy Iteration

- As before, alternate between policy evaluation and policy improvement

$$\pi_0 \xrightarrow{\text{evaluate}} q_{\pi_0} \xrightarrow{\text{improve}} \pi_1 \xrightarrow{\text{evaluate}} q_{\pi_1} \xrightarrow{\text{improve}} \pi_2 \xrightarrow{\text{evaluate}} \cdots$$

- Improve: Given estimate q_{π_k}, update $\pi_{k+1}(s) = \arg\max_a q_{\pi_k}(s, a)$

Monte Carlo Policy Iteration

- As before, alternate between policy evaluation and policy improvement

$$\pi_0 \xrightarrow{\text{evaluate}} q_{\pi_0} \xrightarrow{\text{improve}} \pi_1 \xrightarrow{\text{evaluate}} q_{\pi_1} \xrightarrow{\text{improve}} \pi_2 \xrightarrow{\text{evaluate}} \cdots$$

- Improve: Given estimate q_{π_k}, update $\pi_{k+1}(s) = \arg\max_a q_{\pi_k}(s,a)$
- Evaluate: Estimate q_{π_k} from π_k
  - Iterate over large number of episodes to estimate average values
  - Exploring starts

Need not exhaustively calculate q_{π_k} to update π_k

Like value iteration — update π with partial inf

Monte Carlo Policy Iteration, estimating π_*

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 $Returns(S_t, A_t)$

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

$\pi(S_t) \leftarrow \text{argmax}_a Q(S_t, a)$

ε-soft policies

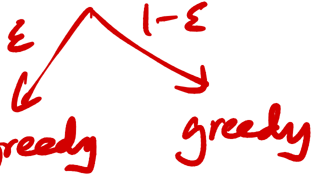
- To avoid exploring starts, use a version of ε greedy

- To avoid exploring starts, use a version of ε greedy
- ε-soft policy
  - Let $\mathcal{A}(s)$ be set of actions available at state s
  - Choose non-greedy action with probability $\frac{\epsilon}{|\mathcal{A}(s)|}$ — uniform
  - Choose greedy action with probability $(1-\varepsilon) + \frac{\epsilon}{|\mathcal{A}(s)|}$

$|\mathcal{A}(s)| = m$ is no. of actions

uniformly choose an action

$\frac{\varepsilon}{m}$ ax



Unique greedy choice ak

1-ε

Monte Carlo Policy Iteration with ε-soft policies

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

Algorithm parameter: small $\varepsilon > 0$

Initialize:

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

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

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

Repeat 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, $t = T-1, T-2, \ldots, 0$:

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

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 $Returns(S_t, A_t)$

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

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

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

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

Off policy methods

- Use a different policy b to generate episodes to estimate v_π

On policy

Our latest estimate of π also generates next episode

Off policy methods

- Use a different policy b to generate episodes to estimate v_π
- Coverage If $\pi(a \mid s) > 0$ then $b(a \mid s) > 0$

Off policy methods

- Use a different policy b to generate episodes to estimate v_π
- Coverage If $\pi(a \mid s) > 0$ then $b(a \mid s) > 0$
- Consider the probability of a trajectory $A_t, S_{t+1}, A_{t+1}, \ldots, S_T$ from S_t

Off policy methods

- Use a different policy b to generate episodes to estimate v_π
- Coverage If $\pi(a \mid s) > 0$ then $b(a \mid s) > 0$
- Consider the probability of a trajectory $A_t, S_{t+1}, A_{t+1}, \ldots, S_T$ from S_t
  - For π, $\pi(A_t \mid S_t)p(S_{t+1} \mid S_t, A_t)\pi(A_{t+1} \mid S_{t+1}) \cdots p(S_T \mid S_{T-1}, A_{T-1})$

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

prob of choosing A_k at S_k acc to π

prob that choosing A_k takes me to S_{k+1}, with R_{k+1}

Off policy methods

- Use a different policy b to generate episodes to estimate v_π
- Coverage If $\pi(a \mid s) > 0$ then $b(a \mid s) > 0$
- Consider the probability of a trajectory $A_t, S_{t+1}, A_{t+1}, \ldots, S_T$ from S_t
  - For π, $\pi(A_t \mid S_t) p(S_{t+1} \mid S_t, A_t) \pi(A_{t+1} \mid S_{t+1}) \cdots p(S_T \mid S_{T-1}, A_{T-1})$
    $$= \prod_{k=t}^{T-1} \pi(A_k \mid S_k) p(S_{k+1} \mid S_k, A_k)$$
  - For b, $b(A_t \mid S_t) p(S_{t+1} \mid S_t, A_t) b(A_{t+1} \mid S_{t+1}) \cdots p(S_T \mid S_{T-1}, A_{T-1})$
    $$= \prod_{k=t}^{T-1} b(A_k \mid S_k) p(S_{k+1} \mid S_k, A_k)$$

Off policy methods

- Use a different policy b to generate episodes to estimate v_π
- Coverage If $\pi(a \mid s) > 0$ then $b(a \mid s) > 0$
- Consider the probability of a trajectory $A_t, S_{t+1}, A_{t+1}, \ldots, S_T$ from S_t
  - For π, $\pi(A_t \mid S_t)p(S_{t+1} \mid S_t, A_t)\pi(A_{t+1} \mid S_{t+1}) \cdots p(S_T \mid S_{T-1}, A_{T-1})$

    $$= \prod_{k=t}^{T-1} \pi(A_k \mid S_k)p(S_{k+1} \mid S_k, A_k)$$
  - For b, $b(A_t \mid S_t)p(S_{t+1} \mid S_t, A_t)b(A_{t+1} \mid S_{t+1}) \cdots p(S_T \mid S_{T-1}, A_{T-1})$

    $$= \prod_{k=t}^{T-1} b(A_k \mid S_k)p(S_{k+1} \mid S_k, A_k)$$
  - $p(s' \mid s, a)$ are unknown model probabilities

Off policy methods

- Use a different policy b to generate episodes to estimate v_π
- Coverage If $\pi(a \mid s) > 0$ then $b(a \mid s) > 0$
- Consider the probability of a trajectory $A_t, S_{t+1}, A_{t+1}, \ldots, S_T$ from S_t
  - For π, $\pi(A_t \mid S_t)p(S_{t+1} \mid S_t, A_t)\pi(A_{t+1} \mid S_{t+1}) \cdots p(S_T \mid S_{T-1}, A_{T-1})$
    $$= \prod_{k=t}^{T-1} \pi(A_k \mid S_k)p(S_{k+1} \mid S_k, A_k)$$
  - For b, $b(A_t \mid S_t)p(S_{t+1} \mid S_t, A_t)b(A_{t+1} \mid S_{t+1}) \cdots p(S_T \mid S_{T-1}, A_{T-1})$
    $$= \prod_{k=t}^{T-1} b(A_k \mid S_k)p(S_{k+1} \mid S_k, A_k)$$
  - $p(s' \mid s, a)$ are unknown model probabilities
  - Take ratio, these cancel out $\dfrac{\prod_{k=t}^{T-1} \pi(A_k \mid S_k)p(S_{k+1} \mid S_k, A_k)}{\prod_{k=t}^{T-1} b(A_k \mid S_k)p(S_{k+1} \mid S_k, A_k)} = \dfrac{\prod_{k=t}^{T-1} \pi(A_k \mid S_k)}{\prod_{k=t}^{T-1} b(A_k \mid S_k)}$

Weighted sampling

- Use ratio $\rho_{t:T} = \frac{\prod_{k=t}^{T-1} \pi(A_k \mid S_k)}{\prod_{k=t}^{T-1} b(A_k \mid S_k)}$ to “adjust” estimates learnt via b
  - Let G_t be an estimate learn by exploring using b
  - The corresponding estimate with respect to π is $\rho_{t:T} G_t$

Weighted sampling

- Use ratio $\rho_{t:T} = \frac{\prod_{k=t}^{T-1} \pi(A_k \mid S_k)}{\prod_{k=t}^{T-1} b(A_k \mid S_k)}$ to "adjust" estimates learnt via b
  - Let G_t be an estimate learn by exploring using b
  - The corresponding estimate with respect to π is $\rho_{t:T} G_t$
- Importance sampling
  - Generate episodes using b
  - Compute adjusted estimates to update q_π, π
  - Contribution of each episode is proportional to its likelihood with respect to π

Weighted sampling

- Use ratio $\rho_{t:T} = \dfrac{\prod_{k=t}^{T-1} \pi(A_k \mid S_k)}{\prod_{k=t}^{T-1} b(A_k \mid S_k)}$ to "adjust" estimates learnt via b
  - Let G_t be an estimate learn by exploring using b
  - The corresponding estimate with respect to π is $\rho_{t:T} G_t$

$$\frac{w_1 G_1 + w_2 G_2}{2}$$

- Importance sampling
  - Generate episodes using b
  - Compute adjusted estimates to update q_π, π
  - Contribution of each episode is proportional to its likelihood with respect to π

$$\frac{w_1 G_1 + w_2 G_2}{w_1 + w_2}$$

- Variations — ordinary importance sampling (above), weighted importance sampling

Weighted sampling

- Use ratio $\rho_{t:T} = \frac{\prod_{k=t}^{T-1} \pi(A_k \mid S_k)}{\prod_{k=t}^{T-1} b(A_k \mid S_k)}$ to "adjust" estimates learnt via b
  - Let G_t be an estimate learn by exploring using b
  - The corresponding estimate with respect to π is $\rho_{t:T} G_t$
- Importance sampling
  - Generate episodes using b
  - Compute adjusted estimates to update q_π, π
  - Contribution of each episode is proportional to its likelihood with respect to π
- Variations — ordinary importance sampling (above), weighted importance sampling
- Off policy methods still to be fully analyzed

Weighted sampling

- Use ratio $\rho_{t:T} = \dfrac{\prod_{k=t}^{T-1} \pi(A_k \mid S_k)}{\prod_{k=t}^{T-1} b(A_k \mid S_k)}$ to "adjust" estimates learnt via b
  - Let G_t be an estimate learn by exploring using b
  - The corresponding estimate with respect to π is $\rho_{t:T} G_t$
- Importance sampling
  - Generate episodes using b
  - Compute adjusted estimates to update q_π, π
  - Contribution of each episode is proportional to its likelihood with respect to π
- Variations — ordinary importance sampling (above), weighted importance sampling
- Off policy methods still to be fully analyzed