Monte Carlo Methods

Wachter Alexander

*Key Concept      * MC Prediction      * MC estimation of action values      * MC control      * ON/OFF - policy methods
Introduction

- Key concept
- Monte Carlo Prediction
- MC estimation of action values
- MC control
- ON/OFF – policy methods
Key Concept

• Not complete knowledge
• Require only experience
• Episodic
• sample and average returns for each state–action
• learn value functions from sample returns with the MDP
• Estimating value of single state independent of #states
MC Prediction

• Learning the state-value function
• Average the returns observed after visits to that state
• $v_\pi(s) =$ value of a state $s$ under policy $\pi$
• Each occurrence of state = visit
  • First-visit MC
  • Every-visit MC
MC Prediction

A+3 → A+2 → B-4 → A+4 → B-3 → terminate
B-2 → A+3 → B-3 → terminate

A+3 → A indicates a transition from state A to state A, with reward of +3

First-Visit

For A:
3+2-4+4-3=2
3-3=0

For B :
-4+4-3 = -3
-2+3-3 = -2

Every Visit

For A:
3+2-4+4-3=2
2-4+4-3= -1
4-3 = 1
3-3=0

For B :
-4+4-3= -3
-3 = -3
-2+3-3 = -2
-3 = -3

V(A) = \frac{1}{2} * (2 + 0) = 1

V(B) = \frac{1}{2} * (-3 - 2) = \frac{-5}{2}

V(A) = \frac{1}{4} * (2 - 1 + 1 + 0) = \frac{1}{2}

V(B) = \frac{1}{4} * (-3 - 3 - 2 - 3) = \frac{-11}{4}
MC Prediction

• Pseudocode:

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

Initialize:
   $\pi \leftarrow$ policy to be evaluated
   $V \leftarrow$ an arbitrary state-value function
   $Returns(s) \leftarrow$ an empty list, for all $s \in S$

Repeat forever:
   Generate an episode using $\pi$
   For each state $s$ appearing in the episode:
      $G \leftarrow$ the return that follows the first occurrence of $s$
      Append $G$ to $Returns(s)$
      $V(s) \leftarrow$ average($Returns(s)$)

Both first-visit MC and every-visit MC converge to $v_\pi(s)$ as the number of visits (or first visits) to $s$ goes to infinity
MC Prediction

Example Blackjack:

• episodic finite MDP (\(\gamma = 1\))
• 200 states
• Policy: hit till sum = 20 or 21
• Goal: find value function for this policy
MC Prediction

After 10,000 episodes

Usable ace

No usable ace

After 500,000 episodes

+1

-1

A

Dealer showing

10

12 Player sum

21
MC Prediction

- Monte Carlo methods do not bootstrap
- Estimating value of single state independent of #states
MC estimation of action values

- If a model is not available, useful to estimate action values
- Goal is to estimate $q^*$
- $q_{\pi}(s, a) =$ expected return when starting in state $s$, taking action $a$ and following policy $\pi$

Complication:
- many state action pairs may never be visited
- general problem of maintaining exploration

Solution:
- exploring starts
- stochastic policy
MC control

- generalized policy iteration at optimal policies (GPI)
  - approximate policy
  - approximate value function

- Simple case:

\[ \pi_0 \xrightarrow{E} q_{\pi_0} \xrightarrow{\pi_1} q_{\pi_1} \xrightarrow{\pi_2} \cdots \xrightarrow{\pi_k} q_k, \]

- Assumptions:
  - Infinite number of episodes $\rightarrow q_{nk}$ exactly
  - Policy improvement is done by making the policy greedy with respect to the current value function
MC control

Policy = with maximal action value:
\[ \pi(s) = \arg \max_a q(s, a). \]

Due to policy improvement theorem
\[ \pi_{k+1} > \pi_k \]
Improvement of \( \pi_{k+1} \) with respect to \( q_{\pi_k} \)

Unlike assumptions :
- Exploring starts
- Infinite number of episodes
MC control

Remove Infinite number of episodes via:
- make steps till error in estimates (requires far too many episodes to be useful)
- give up trying to complete policy evaluation before returning to policy improvement

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

Initialize, for all $s \in S$, $a \in A(s)$:
$$Q(s, a) \leftarrow \text{arbitrary}$$
$$\pi(s) \leftarrow \text{arbitrary}$$
$$Returns(s, a) \leftarrow \text{empty list}$$

Repeat forever:
- Choose $S_0 \in S$ and $A_0 \in A(S_0)$ s.t. all pairs have probability $> 0$
- Generate an episode starting from $S_0, A_0$, following $\pi$
  - For each pair $s, a$ appearing in the episode:
    - $G \leftarrow$ the return that follows the first occurrence of $s, a$
    - Append $G$ to $Returns(s, a)$
    - $Q(s, a) \leftarrow \text{average}(Returns(s, a))$
  - For each $s$ in the episode:
    - $\pi(s) \leftarrow \text{argmax}_a Q(s, a)$
MC control

• Blackjack Revisited:
ON/OFF - policy methods

• Remove: Exploring Starts
  • On-policy methods attempt to evaluate or improve the policy that is used to make decisions, whereas off-policy methods evaluate or improve a policy different from that used to generate the data.

• On-policy learning:
  • „Learn on the job“
  • Learn about policy $\pi$ from experience sampled from $\pi$

• Off-policy learning:
  • „look over someone´s shoulder“
  • Learn about policy $\pi$ from experience sampled from $b$
ON- policy methods

- policy is generally soft $\pi(a|s) > 0$
- Shifts to deterministic optimal policy

epsilon-greedy policies:
- nongreedy actions are given the minimal probability $\frac{\varepsilon}{|A(s)|}$
- Greedy action are given the probability $1 - \varepsilon + \frac{\varepsilon}{|A(s)|}$
- $\varepsilon$ – greedy policy defined as $\pi(a|s) > \frac{\varepsilon}{|A(s)|}$
OFF- policy methods

• ON-policy is a compromise—it learns action values not for the optimal policy, but for a near-optimal policy that still explores.

• Important:
  • Learning from observing humans and other agents
  • Re-use experience generated from old policies

• two policies:
  • Optimal policy (target policy)
  • Exploratory (behavior policy)
OFF- policy methods

• Prediction:
  • both target and behavior policies are fixed
  • Estimate $v_{\pi}$ and $q_{\pi}$, but policy is $b$ (behavior policy)
  • Every action taken under $\pi$ also occasionally in $b$ (coverage)
  • $b$ must be stochastic

• Control:
  • target policy is typically the deterministic greedy policy
  • behavior policy remains stochastic (epsilon-greedy policy)
Summary

• Learn from experience
  • Learn directly from interaction without model
  • Can learn with simulation
  • No bootstrapping

• Need to maintain exploration for Control
  • Exploring starts: unlikely in learning from real experience
  • On-policy: maintain exploration in policy
  • Off-policy: separate behavior and target policies