Monte Carlo methods

Corrado Possieri

Machine and Reinforcement Learning in Control Applications

First learning method

- Unlike the previous lectures, we do not assume knowledge of the environment.
- Monte Carlo methods require only **experience**

$$S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_T, A_T, R_T.$$

- We can learn from *simulated experience*
  - use a model to get experience;
  - often explicit distributions are infeasible.

Monte Carlo methods

- Learning based on averaging sample returns.
- **Model-free**: no knowledge of MDP transitions and rewards.
- We define Monte Carlo methods only for episodic tasks.
- Similar to bandit methods
  - each state is like a different bandit;
  - the different bandit problems are interrelated;
  - the return after taking an action in one state depends on the actions taken in later states in the same episode;
  - the problem becomes non-stationary.

Monte-Carlo Policy Evaluation

- Recall the definition of return

$$G_t = \underbrace{R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{T-t-1} R_T}_{\text{episodic task}}.$$

- Recall the definition of value function

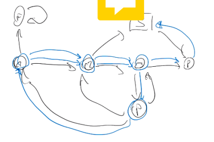
$$v_\pi(s) = \mathbb{E}[G_t | S_t = s]$$

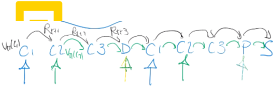
- Monte-Carlo policy evaluation uses empirical mean return instead of expected return.

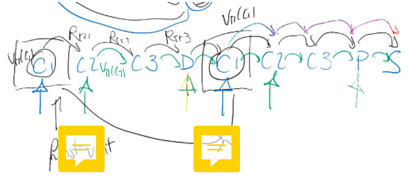
Monte Carlo Prediction

- Given π, we wish to estimate $v_\pi(s)$, given a set of episodes obtained by following π and passing through s.
- Each occurrence of state s in an episode is called a **visit** to s.
- s may be visited multiple times in the same episode
  - the *first-visit MC method* estimates $v_\pi(s)$ as the average of the returns following the first visits to s;
  - the *every-visit MC method* estimates $v_\pi(s)$ as the average of the returns following all the visits to s.

Esempio







First-visit Monte Carlo prediction

First-visit Monte Carlo prediction

Input: policy π
Output: estimate of v_π

Initialization

$V(s) \leftarrow 0, \forall s \in \mathcal{S}$
$N(s) \leftarrow 0, \forall s \in \mathcal{S}$

Loop

generate an episode following π: $S_0, A_0, R_1, S_1, A_1, \ldots, S_{T-1}, A_{T-1}, R_T$
$G \leftarrow 0$
for each step $t = T-1, T-2, \ldots, 0$ **do**
  $G \leftarrow \gamma G + R_{t+1}$
  if S_t does not appear in $S_0, \ldots, S_{t-1}$ **then**
    $N(S_t) \leftarrow N(S_t) + 1$
    $V(S_t) \leftarrow V(S_t) + \dfrac{1}{N(S_t)}(G - V(S_t))$

Every-visit Monte Carlo prediction

Every-visit Monte Carlo prediction

Input: policy π
Output: estimate of v_π

Initialization

$V(s) \leftarrow 0, \forall s \in \mathcal{S}$
$N(s) \leftarrow 0, \forall s \in \mathcal{S}$

Loop

Generate an episode following π: $S_0, A_0, R_1, S_1, A_1, \ldots, S_{T-1}, A_{T-1}, R_T$
$G \leftarrow 0$
for each step $t = T-1, T-2, \ldots, 0$ **do**
$G \leftarrow \gamma G + R_{t+1}$
$N(S_t) \leftarrow N(S_t) + 1$

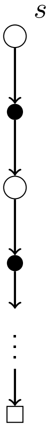
$$V(S_t) \leftarrow V(S_t) + \frac{1}{N(S_t)}(G - V(S_t))$$

Convergence of MC prediction

- In first-visit MC G_t is an independent, identically distributed estimate of $v_\pi(s)$ with finite variance
  - by the law of large numbers $V(s) \to v_\pi(s)$ as $N(s) \to \infty$;
  - the standard deviation falls as $\frac{1}{\sqrt{N}}$.
- Every-visit MC converges quadratically.

Notes on MC prediction

- Does not require probabilities in advance
- Considers only sampled trajectories on one episode.
- The estimates for each state are independent
  - it does not bootstrap.
- Computational expense is independent of the number of states.



Monte Carlo Estimation of Action Values

- With a model, state values are sufficient to determine a policy.
- If a model is not available, it would be better to estimate q_*
  - $\pi_*(s) = \arg\min_a q_*(s,a)$.
- Recall that $q_\pi(s,a)$ is the expected return when starting in state s, taking action a, and thereafter following policy π.
- Monte Carlo methods can be used to estimate q_π
  - we visit state–action pairs rather than states;
  - pair s,a is visited in an episode if state s is visited and action a is taken.
  - we still have first-visit and every-visit methods.

The importance of exploration

- Many state–action pairs may never be visited
  - following π we observe returns only for pairs $s, \pi(s)$;
  - Monte Carlo estimates of the other actions will not improve.
- We need to *maintain exploration*
  - episodes start at a given state-action pair;
  - every pair has a nonzero probability of being selected ;
  - this is usually referred to as **exploring starts**.
- Another approach relies on stochastic policies with nonzero exploring probability.

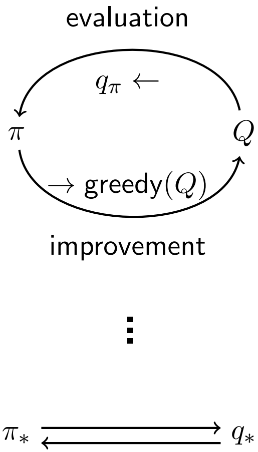
Monte Carlo control

- Use the same idea of GPI.
- Alternate evaluation and improvement

$$\pi_0 \xrightarrow{\mathrm{E}} q_{\pi_0} \xrightarrow{\mathrm{I}} \pi_1 \xrightarrow{\mathrm{E}} q_{\pi_1} \xrightarrow{\mathrm{I}} \pi_2 \xrightarrow{\mathrm{E}} q_{\pi_2} \xrightarrow{\mathrm{I}} \dots.$$

- Evaluation carried out via MC prediction.
- Greedy policy improvement

$$\pi(s) \leftarrow \max_a q_\pi(s,a).$$



Convergence of Monte Carlo control

- Assume that
  - we observed an infinite number of episodes;
  - episodes are initialized with exploring start.
- The policy improvement theorem applies

$$
\begin{aligned}
q_{\pi_k}(s, \pi_{k+1}(s)) &= q_{\pi_k}(s, \arg\max_a q_{\pi_k}(s,a)) \\
&= \max_a q_{\pi_k}(s,a) \geqslant q_{\pi_k}(s, \pi_k(s)) = v_{\pi_k}(s).
\end{aligned}
$$

  - $\pi' \geqslant \pi$;
  - $\pi' = \pi \implies$ both policies are optimal.

Removing infinite episodes hypothesis

- We assumed that policy evaluation operates on an infinite number of episodes to guarantee that $Q \leftarrow q_\pi$.
- In VI, we already noticed that this is not necessary
  - policy evaluation between each step of policy improvement.
- Alternate between evaluation and improvement for states.

Monte Carlo exploring start

Monte Carlo exploring start

Output: estimate of π_*

Initialization

$Q(s,a) \leftarrow 0, \forall s \in \mathcal{S}, \forall a \in \mathcal{A}$
$N(s,a) \leftarrow 0, \forall s \in \mathcal{S}, \forall a \in \mathcal{A}$
$\pi(s) \leftarrow \text{random}, \forall s \in \mathcal{S}$

Loop

chose S_0, A_0 randomly so that all pairs have nonzero probability
generate an episode following π: $S_0, A_0, R_1, S_1, A_1, \ldots, S_{T-1}, A_{T-1}, R_T$
$G \leftarrow 0$
for each step $t = T-1, T-2, \ldots, 0$ **do**
  $G \leftarrow \gamma G + R_{t+1}$
  if S_t, A_t does not appear in $S_0, A_0 \ldots, S_{t-1}, A_{t-1}$ **then**
    $N(S_t, A_t) \leftarrow N(S_t, A_t) + 1$
    $$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{1}{N(S_t, A_t)} (G - Q(S_t, A_t))$$
    $\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$

On-policy vs off-policy

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.

ε-soft policies

- In on-policy control methods the policy is generally *soft*
  - $\pi(a|s) > 0$, $\forall a \in \mathcal{A}(s), \forall s \in \mathcal{S}$.
- ε-soft policies satisfy $\pi(a|s) \geqslant \frac{\varepsilon}{|\mathcal{A}(s)|}$, $\forall a \in \mathcal{A}(s), \forall s \in \mathcal{S}$.
- ε-greedy policies

$$\pi(a|s) = \begin{cases} 1 - \varepsilon + \frac{\varepsilon}{|\mathcal{A}(s)|}, & \text{if } a = \arg\max_a q(s, a), \\ \frac{\varepsilon}{|\mathcal{A}(s)|}, & \text{otherwise,} \end{cases}$$

azione ottima

Azione casuale

are examples of ε-soft policies.

- To preserve exploration
  - move policy to an ε-greedy one.

Removing exploring start

On-policy first-visit Monte Carlo control

Input: $\varepsilon > 0$
Output: estimate of π_*

Initialization

$Q(s,a) \leftarrow 0, \forall s \in \mathcal{S}, \forall a \in \mathcal{A}$
$N(s,a) \leftarrow 0, \forall s \in \mathcal{S}, \forall a \in \mathcal{A}$
$\pi(s) \leftarrow$ arbitrary ε-soft policy, $\forall s \in \mathcal{S}$

Loop

generate an episode following π: $S_0, A_0, R_1, S_1, A_1, \ldots, S_{T-1}, A_{T-1}, R_T$
$G \leftarrow 0$
for each step $t = T-1, T-2, \ldots, 0$ **do**
  $G \leftarrow \gamma G + R_{t+1}$
  if S_t, A_t does not appear in $S_0, A_0 \ldots, S_{t-1}, A_{t-1}$ **then**
    $N(S_t, A_t) \leftarrow N(S_t, A_t) + 1$

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{1}{N(S_t, A_t)} (G - Q(S_t, A_t))$$

    $A^* \leftarrow \arg\max_a Q(S_t, a)$
    for all $a \in \mathcal{A}(S_t)$ **do**

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

Si utilizza una policy stocastica e-greedy. La stima della funzione di qualità rimane invariata nei metodi di Monte Carlo.
L'aggiornamento della policy deve essere stocastico per mantenere l'esplorazione: si aggiorna seconda la regola della slide 17, cioè e-greedy rispetto alla funzione q(s,a) di qualità.

Policy improvement theorem for ε-greedy policies

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

- By the policy improvement theorem
  - $\pi' \geqslant \pi$.

Se si passa da una vecchia policy stocastica di tipo e-soft ad una e-greedy, la nuova policy è migliore delle vecchia. Così si può implementare l'iterazione di valutazione e miglioramento per convergere alla funzione di qualità ottima e alla policy ottima.

Modified environment for ε-soft policies

- Consider a modified environment that behaves as follows
  - if in state s and taking action a, then with probability $1 - \varepsilon$ the new environment behaves like the old one;
  - with probability ε it repicks the action at random, with equal probabilities.
- The best one can do in this new environment with deterministic policies is the same as the best one could do in the original environment with ε-soft policies.
- Let $\tilde{v}_*$ and $\tilde{q}_*$ be the optimal value functions in the new environment.
- π is optimal among ε-soft policies if and only if $v_\pi = \tilde{v}_*$.

Optimal ε-soft policies

- In the new environment Bellman equation reads as

Ambiente modificato

$$\tilde{v}_*(s) = (1-\varepsilon)\max_a q_*(s,a) + \frac{\varepsilon}{|\mathcal{A}(S_t)|}\sum_a q_*(s,a)$$
$$= (1-\varepsilon)\max_a \sum_{s',r} p(s',r|s,a)\left(r+\gamma\tilde{v}_*(s')\right)$$
$$+ \frac{\varepsilon}{|\mathcal{A}(S_t)|}\sum_{a,s',r} p(s',r|s,a)\left(r+\gamma\tilde{v}_*(s')\right)$$

On-policy sfrutto la policy in ogni istante di tempo per generare l'esperienza per capire quale è quella ottima.

- On the other hand, if v_π is no longer improved

$$v_\pi(s) = \frac{\varepsilon}{|\mathcal{A}(S_t)|}\sum_a q_\pi(s,a) + (1-\varepsilon)\max_a q_\pi(s,a)$$
$$= (1-\varepsilon)\max_a \sum_{s',r} p(s',r|s,a)\left(r+\gamma v_\pi(s')\right)$$
$$+ \frac{\varepsilon}{|\mathcal{A}(S_t)|}\sum_{a,s',r} p(s',r|s,a)\left(r+\gamma v_\pi(s')\right)$$

N.B.: $\tilde{v}_*(s) = v_\pi(s)$

- $\tilde{v}_*$ is unique
  - $\pi' = \pi \implies \pi$ is the optimal ε-soft policy.

Off-policy methods

Qualcuno prende le decisione d io devo predirne il comportamento futurp Ho solo informazioni di azione,stato,reward

- On-policy methods learn action values not for the optimal policy, but for a near-optimal policy that explores.

- We can also think of using two policies

  target policy: policy that is learned; coincide con la policy ottim. Si può ipotizzare che questa sia di tipo deterministico

  behavior policy: policy used to learn. policy utilizzata per apprendere;potrebbe essere una policy qualsiasi

- Off-policy methods
  - are more general;
  - are more complex;
  - are slower to converge;
  - can be used to learn from data;
  - learn about optimal policy while following exploratory policy;
  - learn about multiple policies while following one policy;
  - reuse previous experience.

Non si impara la policy ottima, ma imparano una policy vicino a quella ottima, ma che mantengono un grado di stocasticità per mantenere l'esplorazione.
target e behavior possono essere anche uguali.
Sono metodi più generali rispetto a quelli off-policy.
Si ha una convergenza più lenta.

Off-policy prediction

- We want to estimate v_π (or q_π)
- The target policy is π
  - might be deterministic.
- The behavior policy is b
  - might be stochastic;
  - aimed at exploration.
- To learn π using b, we need the *coverage* assumption

$$\pi(a|s) > 0 \implies b(a|s) > 0.$$

se la policy assegna all'azione a un probabilità >0, allora deve fare la stessa cosa anche la policy di comportamento.

Importance sampling

Vogliamo stimare la funzione valore e qualità quanso di segue la policy pi.

- Estimate expected values under one distribution given samples from another.

policy di comportamento

- Weighting returns according to the relative probability of their trajectories occurring under the target and behavior policies.
- Given S_t and π

Probabilità indipendenti

$$\mathbb{P}\left[A_t, S_{t+1}, A_{t+1}, \ldots, S_T | S_t, A_{t:T-1} \sim \pi\right] = \prod_{k=t}^{T-1} \pi(A_k|S_k)p(S_{k+1}|S_k, A_k).$$

Coincide con la probabilità dell'episodio

- The importance-sampling ratio is

Non dipende dalla conoscenza del modello, ma devo conoscere solamente la policy target e di comportamento.

$$\rho_{t:T-1} = \frac{\prod_{k=t}^{T-1} \pi(A_k|S_k)p(S_{k+1}|S_k, A_k)}{\prod_{k=t}^{T-1} b(A_k|S_k)p(S_{k+1}|S_k, A_k)} = \underbrace{\frac{\prod_{k=t}^{T-1} \pi(A_k|S_k)}{\prod_{k=t}^{T-1} b(A_k|S_k)}}_{\text{depends only on } \pi \text{ and } b}.$$

Off-policy expectation

Se mi calcolassi la funzione valore della policy di comportamento è diverso dalla funzione valore della policy target

- Returns have wrong expectation

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

- The importance sampling ratio transforms the returns to have the right expected value

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

Off-policy Monte Carlo prediction

- Let
  - $\mathcal{T}(s)$: set of all time steps in which state s is visited;
  - $T(t)$: first time of termination following time t;
  - G_t: return after t up to $T(t)$.
- Ordinary importance sampling

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

- Weighted importance sampling

$$V(s) = \begin{cases} \frac{\sum_{t\in\mathcal{T}(s)} \rho_{t:T(t)-1} G_t}{\sum_{t\in\mathcal{T}(s)} \rho_{t:T(t)-1}}, & \text{if } \sum_{t\in\mathcal{T}(s)} \rho_{t:T(t)-1} > 0, \\ 0, & \text{otherwise.} \end{cases}$$

Comparison of importance sampling

Ordinary	Weighted
• Unbiased.	• Biased $\to 0$.
• Unbounded variance.	• Bounded variance $\to 0$.

- There are other classes of importance sampling
  - discounting-aware importance sampling;
  - per-decision importance sampling.
- Rather technical (see more on textbook).

Importance sampling for state-action value functions

- Given S_t, A_t, and π

$$\begin{aligned}&\mathbb{P}\left[A_t, S_{t+1}, A_{t+1}, \ldots, S_T | S_t, A_t, A_{t+1:T-1} \sim \pi\right] \\ &= p(S_{t+1}|S_t, A_t)\pi(A_{t+1}|S_{t+1})p(S_{t+2}|S_{t+1}, A_{t+1}), \ldots, p(S_T|S_{T-1}, A_{T-1}) \\ &= p(S_{t+1}|S_t, A_t) \prod_{k=t+1}^{T-1} \pi(A_k|S_k)p(S_{k+1}|S_k, A_k).\end{aligned}$$

- The importance-sampling ratio is

$$\varrho_{t:T-1} = \frac{p(S_{t+1}|S_t, A_t) \prod_{k=t+1}^{T-1} \pi(A_k|S_k)p(S_{k+1}|S_k, A_k)}{p(S_{t+1}|S_t, A_t) \prod_{k=t+1}^{T-1} b(A_k|S_k)p(S_{k+1}|S_k, A_k)} = \underbrace{\frac{\prod_{k=t+1}^{T-1} \pi(A_k|S_k)}{\prod_{k=t+1}^{T-1} b(A_k|S_k)}}_{\text{depends only on } \pi \text{ and } b}.$$

- Weighted importance sampling

$$Q(s,a) = \begin{cases} \dfrac{\sum_{t\in\mathcal{T}(s)} \varrho_{t:T(t)-1} G_t}{\sum_{t\in\mathcal{T}(s)} \varrho_{t:T(t)-1}}, & \text{if } \sum_{t\in\mathcal{T}(s)} \varrho_{t:T(t)-1} > 0, \\ 0, & \text{otherwise.} \end{cases}$$

Incremental implementation of weighted average

- Suppose we want to compute

$$V = \frac{\sum_{k=1}^{n-1} W_k G_k}{\sum_{k=1}^{n-1} W_k}$$

and keep it up-to-date as we obtain a single additional return.

- It suffices to keep track of increments

$$\begin{aligned} C_{n+1} &= C_n + W_{n+1}, \\ V_{n+1} &= V_n + \frac{W_n}{C_n}(G_n - V_n). \end{aligned}$$

with $C_0 = 0$ and V_1 arbitrary.

Off-policy Monte Carlo prediction

Off-policy Monte Carlo prediction

Input: policy π
Output: estimate of q_π

Initialization

$Q(s,a) \leftarrow$ arbitrary, $\forall s \in \mathcal{S}$
$C(s,a) \leftarrow 0, \forall s \in \mathcal{S}$

Loop

$b \leftarrow$any policy with coverage of π
generate an episode following b: $S_0, A_0, R_1, S_1, A_1, \ldots, S_{T-1}, A_{T-1}, R_T$
$G \leftarrow 0$
$W \leftarrow 1$
for each step $t = T-1, T-2, \ldots, 0$ **do**

$G \leftarrow \gamma G + R_{t+1}$
$C(S_t, A_t) \leftarrow C(S_t, A_t) + W$

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)} \left(G - Q(S_t, A_t)\right)$$

$$W \leftarrow W \frac{\pi(A_t|S_t)}{b(A_t|S_t)}$$ Importance sampling

Off-policy Monte Carlo control

- The target policy is the greedy policy with respect to Q.

Stima funzione qualità

- The behavior policy b can be anything
  - choosing b to be ε-soft ensures exploration.

IN caso contrario tutti gli stati non sono sempre visitati e non si mantiene un grado di esplorazione. Per essere sicuro che tutte lo coppie stato azione vengno scelte infinite volte in un certo stato

- Learns only from the tails of episodes with greedy actions.

Off-policy Monte Carlo control

Off-policy Monte Carlo control

Output: π_*

Initialization

$Q(s,a) \leftarrow$ arbitrary, $\forall s \in \mathcal{S}$
$C(s,a) \leftarrow 0, \forall s \in \mathcal{S}$
$\pi(s) \leftarrow \arg\max_a Q(s,a), \forall s \in \mathcal{S}$

Loop

$b \leftarrow$any soft policy
generate an episode following b: $S_0, A_0, R_1, S_1, A_1, \ldots, S_{T-1}, A_{T-1}, R_T$
$G \leftarrow 0$
$W \leftarrow 1$
for each step $t = T-1, T-2, \ldots, 0$ **do**
    $G \leftarrow \gamma G + R_{t+1}$
    $C(S_t, A_t) \leftarrow C(S_t, A_t) + W$
    $$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)} (G - Q(S_t, A_t))$$
    $\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$
    if $A_t \neq \pi(S_t)$ **then**
        proceed to next episode
    else
        $$W \leftarrow W \frac{1}{b(A_t|S_t)}$$