Monte Carlo Methods

M. Heinzer, E. Profumo

11 April 2016

1 Notations and Setting.

1. We use the theoritical framework of **Markov Decision Processes**(MDP) to describe the game evolution. **Episodes** are just different games. Denote by

   - $t \in \{1, 2, ...T_i\}$ describes the different **steps** of the episode i (we will drop i for clarity).
   - $(S_t)_{t\in\{1,2,...T\}}$ the process of different **states of the game**.
   - $(S_t, a_t)_{t\in\{1,2,...T\}}$ the **state-action** pairs. *The actions which can be taken depend on the current state*
   - $(R_t)_{t\in\{1,2,...T\}}$ the process of **rewards** following a triple (state,action,resulting state).

2. MDP's are about an **agent** taking decision in an **environment**.

   It is formalized by a **policy function** π
   $\pi(a|s)$ is the probability of taking action a being in state s.

3. **State value function:**

$$v_\pi(s) = \mathbb{E}_\pi\left(\sum_{i=0}^{\infty} R_{t+i+1} \middle| S_t = s\right)$$

4. **Action-State value function:**

$$q_\pi(s, a) = \mathbb{E}_\pi\left(\sum_{i=0}^{\infty} R_{t+i+1} \middle| S_t = s, A_t = a\right)$$

5. **Policy improvement theorem**

   Let π, π' be deterministic policies on the same environment, then if for all states s

$$q_\pi(s, \pi'(s)) \geq v_\pi(s)$$

   We have that for all $s \in S$ $v_{\pi'}(s) \geq v_\pi(s)$ and so $\pi' \geq \pi$.

The main idea in Monte Carlo methods is that instead of looking into the **complicated probabilistic behaviour** of the environment we just **learn from experience**, from our successes and mistakes.

2 Monte Carlo prediction: Estimation of the value functions.

We estimate the value functions just by recording the gain following the visit to some state s or state-action pair (s, a) and take the **average**.

Suppose we have n episodes and let $N(s)$ be the enumeration of episodes which visited s.

Then we define $\hat{v}_\pi(s)$ as follows:

$$\hat{v}_\pi(s) = \frac{1}{|N(s)|} \sum_{i=1}^{n} R_i I_{i \in N(s)}$$

Each return is an i.i.d. estimate of the true value of $v_\pi(s)$.

Similarly for the Action-state value function

$$\hat{q}_\pi(s, a) = \frac{1}{|N(s, a)|} \sum_{i=1}^{n} R_i I_{i \in N(s,a)}$$

Assumptions to maintain exploration:

- Every pair has non-zero probability of being selected as start. We call this **Exploring Starts**.
- Use **stochastic policies** which have a non-zero probability of selecting all available actions in each state.

  Here is the corresponding algorithm

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

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

3 Monte Carlo control: Improvement of the policies

We use the Generalized Policy Iteration.

We start with an arbitrary policy π_0. At each step we evaluate the state-action function q_{π_i} with Monte Carlo prediction, and select as new policy π_{i+1} the greedy policy corresponding to q_{π_i}:

$$\pi_{i+1}(s) = \arg\max_a q_i(s,a)$$

3.1 How to obtain better convergence result to the optimal policy

We **update the policy after each episode** instead of waiting for many episodes.

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

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

3.2 How to remove exploring starts assumption?

3.2.1 Use soft policies

Soft policy is a policy π such that for all state s and action $a \in A(s)$, $\pi(a|s) > 0$

For example an ϵ-greedy policy instead of the greedy one in Monte Carlo control algorithm:

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

With this way of improving policy it can be shown that we still converge to the optimal policy.

3.2.2 With off-policy methods.

Off-policy learning method is a way of learning the value functions of a policy **via samples generated from another policy.**
This way we can generate samples maintaining exploration and still get the right value function.

- π the target policy
- μ the behavior policy(from which we sample from)

Coverage assumption: $\pi(a, s) > 0 \Rightarrow \mu(a, s) > 0$

We define the relative probability of the trajectory under the target and behavior policies

$$\rho_1^T = \frac{\prod_{k=1}^{T-1} \pi(A_k|S_k)p(S_{k+1}|S_k, A_k)}{\prod_{k=1}^{T-1} \mu(A_k|S_k)p(S_{k+1}|S_k, A_k)} = \prod_{k=1}^{T-1} \frac{\pi(A_k|S_k)}{\mu(A_k|S_k)}$$

Suppose we have gathered experience in the form of n episodes.
Let $N(s)$ be the enumeration of episodes which visited state s.
Let $T(s, k)$ be the first time when state s is visited in episode k. The time of the terminal state of episode k is denoted as $T(k)$.

Our estimates for the target policy state value function is then:

$$\hat{v}_\pi(s) = \frac{\sum_{i \in N(s)} \rho_{T(s,i)}^{T(i)} G_i}{|N(s)|}$$

This is what we call ordinary importance sampling.

$$\hat{v}_\pi(s) = \frac{\sum_{i \in N(s)} \rho_{N(s,i)}^{T(i)} G_i}{\sum_{i \in N(s)} \rho_{N(s,i)}^{T(i)}}$$

This is what we call weighted importance sampling.

Main idea: Instead of taking the usual average we give **more weight** to the events that are **more likely to occur under π.**

References

[1] Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning: An Introduction.* 2nd Edition, 2014,2015.