Model Free Control : Monte Carlo Methods

Easwar Subramanian

TCS Innovation Labs, Hyderabad

Email : easwar.subramanian@tcs.com / cs5500.2020@iith.ac.in

September 30, 2021

Overview

1. Review
2. Towards Model Free Control
3. Policy Evaluation : Action Value Function
4. Monte Carlo Control
5. Exploration

Review

Policy Evaluation from Samples: A Recap

TCS Research & Innovation

$$
\begin{aligned}
V^{\pi}(s) &\stackrel{\text{def}}{=} \mathbb{E}_{\pi}(G_t | s_t = s) \\
&= \mathbb{E}_{\pi}\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\right) \\
&= \mathbb{E}_{\pi}\left[r_{t+1} + \gamma V^{\pi}(s_{t+1}) | s_t = s\right]
\end{aligned}
$$

- Estimate expectation from experience;
  - ★ Using total discounted reward (MC)
  - ★ Using the recursive decomposition formulation of the value function (TD)

TATA CONSULTANCY SERVICES

Eligibility Traces

- The eligibility trace of a state $s \in \mathcal{S}$ at time t is defined recursively by

$$
\begin{aligned}
e_0(s) &= 0 \\
e_t(s) &= \begin{cases} (\lambda\gamma)e_{t-1}(s), & s_t \neq s \\ (\lambda\gamma)e_{t-1}(s) + 1, & s_t = s \end{cases}
\end{aligned}
$$

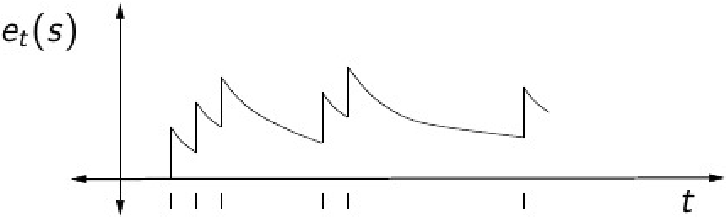


Figure Source: Sutton and Barto

Algorithm : TD(λ)

Algorithm TD(λ) : Algorithm

Initialize $e(s) = 0$ for all s, $V(s)$ arbitrarily
for For each episode **do**
    Let s be a start state for episode k
    for For each step of the episode **do**
        Take action a recommended by policy π from state s
        Collect reward r and reach next state s'
        Form the one-step TD error $\delta \leftarrow r + \gamma V(s') - V(s)$
        Increment eligibility trace of state s, $e(s) \leftarrow e(s) + 1$
        for For all states $S \in \mathcal{S}$ **do**
            Update $V(S)$: $V(S) \leftarrow V(S) + \alpha e(S)\delta$
            Update eligibility trace: $e(S) \leftarrow \lambda\gamma e(S)$
        end for
        Move to next state: $s \leftarrow s'$
    end for
end for

Towards Model Free Control

Problem and Motivation

- **Goal** : How can we learn a good policy?
- **Motivation** : Many real world applications can be modelled as MDP
  - ★ Games like Backgammon and Go
  - ★ Robot Locomotion
  - ★ Inventory or supply chain management
- For almost all these problems, model is unknown or computationally infeasible; but sampling experiences is possible
- Learning better policies through experiences is model free control

Towards Model Free Control

DP algorithms for control

- Value Iteration
- Policy Iteration

Question : How can we do model free control ?

- Value iteration may not come in handy because it requires knowledge of model; so not suitable

$$V_{k+1}(s) \leftarrow \max_a \left[\sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma V_k(s') \right) \right]$$

- How about policy iteration (PI) ?

PI is a two step provess

★ Policy evaluation
★ Policy improvement

Policy Iteration : Recap

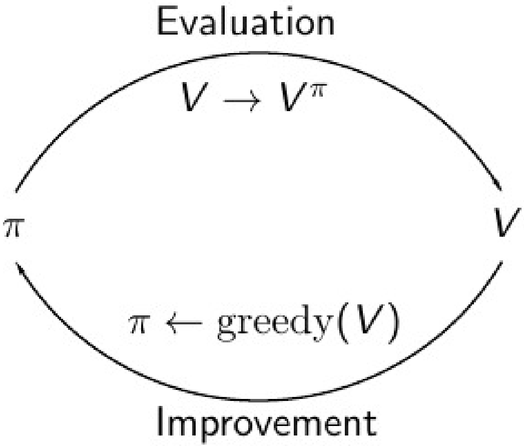


Figure Source: Sutton and Barto

On Policy Improvement From Samples

- (Greedy) Policy improvement

$$\pi(s) \quad = \quad \arg\max_a \sum_{s'} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma V^\pi(s')\right]$$

- Generally, model free control is not done with V as greedy policy improvement over V requires the <u>knowledge of the model</u>
- (Greedy) policy improvement over Q is <u>model free</u>

$$\pi(s) = \arg\max_a Q^\pi(s,a)$$

- For model-free policy improvement, we use Q^π, not V^π

Core Idea behind Model Free Control

- Initialize a policy π
- Repeat
  - Policy Evaluation : Find Q^{π}
  - Policy Improvement : Get an improved policy from evaluation of Q^{π}

Policy Evaluation : Action Value Function

Policy Evaluation : Action Value Function

- We now need to evaluate Q^π instead of V^π
- Recall that the state-action value function of a policy π is given by,

$$
\begin{aligned}
Q^\pi(s,a) &\stackrel{\text{def}}{=} \mathbb{E}_\pi(G_t | s_t = s, a_t = a) \\
&= \mathbb{E}_\pi\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a\right) \\
&= \mathbb{E}_\pi(r_{t+1} + \gamma Q^\pi(s_{t+1}, a_{t+1}) | s_t = s, a_t = a)
\end{aligned}
$$

- We can use MC or TD methods to evaluate Q^π using samples

First Visit Monte Carlo : Action Value Function

- To evaluate $Q^\pi(s, a)$ for some given state s and action a, repeat over several episodes
  - ★ The first time t that $s_t = s$ and $\pi(s) = a$ in the episode
    1. Increment counter for number of visits to s: $N(s, a) \leftarrow N(s, a) + 1$
    2. Increment running sum of total returns with return from current episode: $S(s, a) \leftarrow S(s, a) + G_t$
- Monte Carlo estimate of value function $Q(s, a) \leftarrow S(s, a)/N(s, a)$

The main drawback of this algorithm is

- Many state action pairs may never be visited
- If policy π is deterministic, things get even worse

Exploring Starts Assumption

Exploring Starts (ES) Assumption

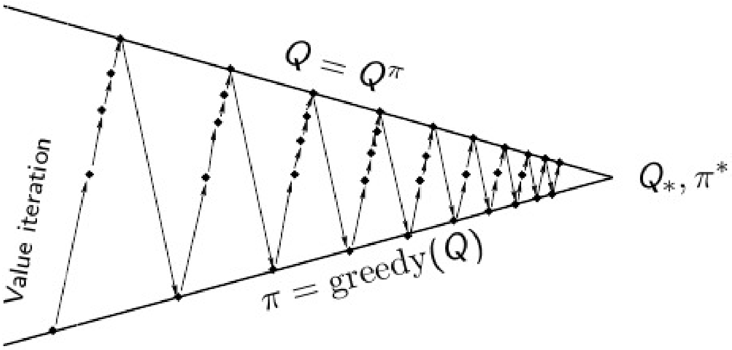
- First step of each episode start at a state-action pair, and that every such pair has non-zero probability of being selected at start
- Guarantees that all state-action pairs will be visited an infinite number of times in the limit of an infinite number of episodes

Not a realistic assumption at all !! But let's assume it for a while

- With ES assumption, first or every visit MC algorithm will evaluate Q^{π}

Monte Carlo Control

Policy Iteration with Action Value Function



- Monte Carlo Policy Evaluation, $Q = Q^{\pi}$
- Greedy policy improvement , $\pi' = \arg\max_a Q^{\pi}(s, a)$

 Figure Source: Sutton and Barto

Monte Carlo Control with ES

TCS Research & Innovation

Algorithm Monte Carlo Control with ES

1: Start with an initial policy π_1;
2: **for** $k = 1, 2, \cdots, K$ **do**
3: Policy Evaluation Step : Evaluate Q^{π_k} using first or every visit MC
4: Policy Improvement Step :

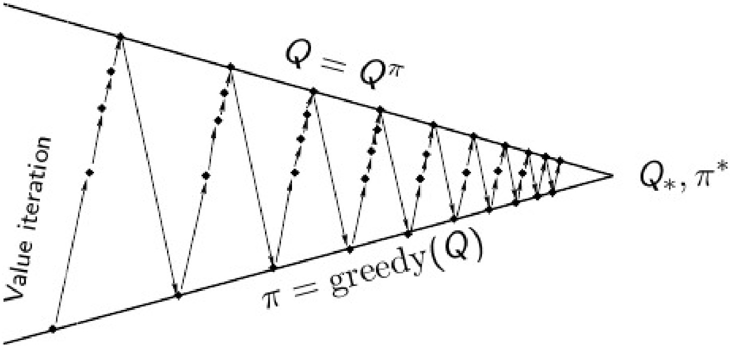
$$\pi_{k+1} = \arg\max_a Q^{\pi_k}(s, a)$$

5: **end for**

- Convergence of policy evaluation to Q^π is assured only under the ES assumption
- Once ES assumption is made, to understand convergence to Q_* and π_* one can use the same kind of arguments as we had in the policy iteration algorithm in the DP setting

TATA CONSULTANCY SERVICES

Policy Iteration with Action Value Function



- Is it good to be always greedy ?
- Should we patiently wait until policy evaluation step converges ?

Exploration

On Greedy Action Selection

- There are two doors in front of you
- You open the left door and get reward 0 i.e. $V(\text{left}) = 0$
- You open the right door and get reward 1 $V(\text{right}) = 1$
- You open the right door and get reward 3 $V(\text{right}) = 2$
- You open the right door and get reward 2 $V(\text{right}) = 3$
- Are we sure that right door is the best door ?

ε-Greedy Exploration

- Simplest idea for ensuring continual exploration
- All m actions are tried with non-zero probability every time
  - With probability $1-\varepsilon$, choose the greedy action
  - With probability ε, choose an action uniformly at random

$$
\begin{aligned}
\pi(a|s) &= \frac{\varepsilon}{m} + 1 - \varepsilon, \text{ if } \; a = \arg\max_{a'} Q(s, a'), \\
&= \frac{\varepsilon}{m}, \text{ otherwise}
\end{aligned}
$$

ε-Greedy Policy Improvement

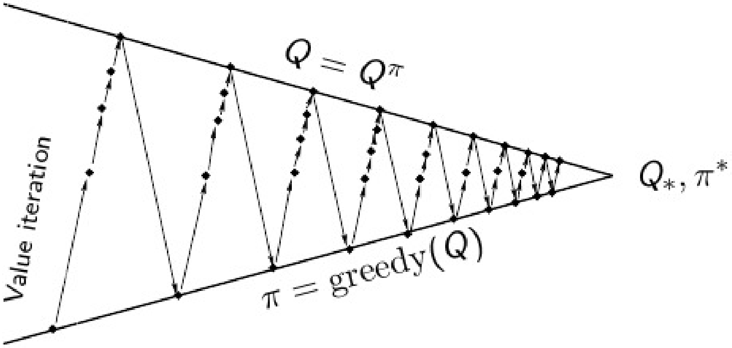
For any policy ε-greedy policy π, the ε-greedy policy π' w.r.t. Q^π is an improvement over π, that is, $V^{\pi'}(s) \geq V^\pi(s)$

$\varepsilon-$ Greedy Policy Improvement

$$
\begin{aligned}
Q^{\pi}(s, \pi'(s)) &= \sum_{a \in \mathcal{A}} \pi'(a|s) Q^{\pi}(s, a) \\
&= \frac{\varepsilon}{m} \sum_{a \in \mathcal{A}} Q^{\pi}(s, a) + (1 - \varepsilon) \max_{a} Q^{\pi}(s, a) \\
&= \frac{\varepsilon}{m} \sum_{a \in \mathcal{A}} Q^{\pi}(s, a) + (1 - \varepsilon) \frac{1 - \varepsilon}{1 - \varepsilon} \max_{a} Q^{\pi}(s, a) \\
&= \frac{\varepsilon}{m} \sum_{a \in \mathcal{A}} Q^{\pi}(s, a) + (1 - \varepsilon) \sum_{a} \frac{\pi(a|s) - \frac{\varepsilon}{m}}{1 - \varepsilon} \max_{a} Q^{\pi}(s, a) \\
&\geq \frac{\varepsilon}{m} \sum_{a \in \mathcal{A}} Q^{\pi}(s, a) + (1 - \varepsilon) \sum_{a} \frac{\pi(a|s) - \frac{\varepsilon}{m}}{1 - \varepsilon} Q^{\pi}(s, a) \\
&= \sum_{a \in \mathcal{A}} \pi(a|s) Q^{\pi}(s, a) = V^{\pi}(s)
\end{aligned}
\tag{1}
$$

Therefore, $V^{\pi'}(s) \geq V^{\pi}(s)$ from the policy improvement theorem

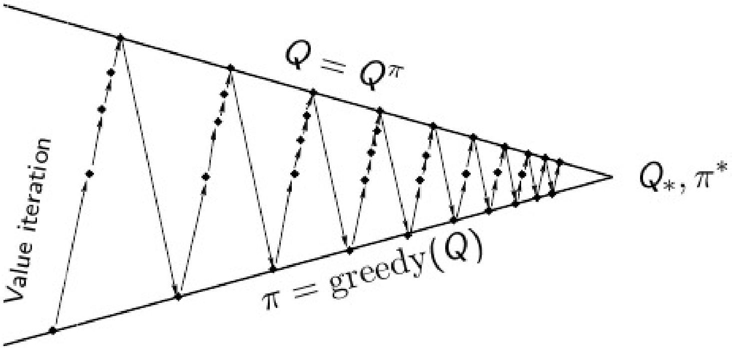
Policy Iteration with Action Value Function



- Monte Carlo Policy Evaluation, $Q = Q^\pi$
- ε-Greedy policy improvement

 Figure Source: Sutton and Barto

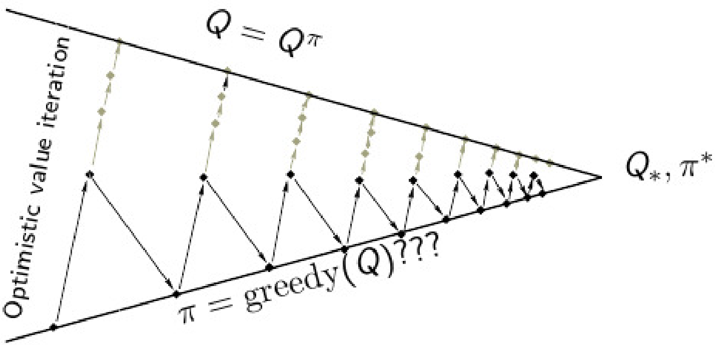
Policy Iteration Revisited



- Should we patiently wait until policy evaluation step converges ?

Figure Source: Sutton and Barto

Optimistic Policy Iteration



We can cut short the evaluation process !

- Monte Carlo Policy Evaluation, $Q \approx Q^{\pi}$
- ε-Greedy policy improvement

Figure Source: Sutton and Barto

GLIE

Definition

G*reedy in the* **L***imit with* **I***nfinite* **E***xploration*

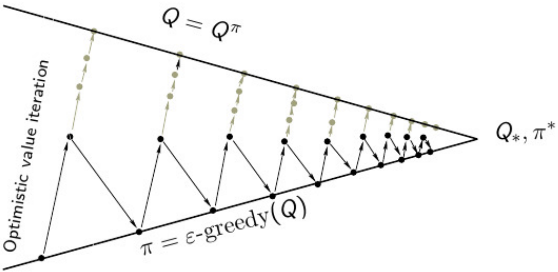
- All state-action pairs are visited infinitely often
- The policy converges to a purely greedy policy

$$\lim_{k \to \infty} \pi_k(a|s) = \mathbf{1}_{a = \arg\max_{a'} Q_k(s,a)}$$

- ε-greedy is GLIE if ε decays to 0 asymptotically, for example,

$$\varepsilon_k = \frac{1}{k}$$

Optimistic GLIE Policy Iteration



Every episode

- Monte Carlo Policy Evaluation $Q \approx Q^\pi$
- Policy improvement using ϵ- greedy with ε decay

Figure Source: Sutton and Barto

GLIE Monte Carlo Control

Algorithm Monte Carlo Control : GLIE

Initalize Q(s,a) = 0, set $\varepsilon = 1$;
Create an ε-greedy initial policy π_1;
for $k = 1, 2, \cdots, K$ **do**
    Sample a trajectory from policy π_k
    for For each state action (s_t, a_t) pair in the trajectory **do**
        Compute the total discounted return G_t starting from (s_t, a_t)

$$N(s_t, a_t) = N(s_t, a_t) + 1$$

$$Q(s_t, a_t) = Q(s_t, a_t) + \frac{1}{N(s_t, a_t)}(G_t - Q(s_t, a_t))$$

    end for
    Set $\epsilon \leftarrow \frac{1}{k}$ and perform the policy improvement step as

$$\pi_{k+1} = \epsilon\text{-greedy}(\pi_k)$$

end for

Monte Carlo Control

- Model free control algorithms interleave policy evaluation and policy improvement
- GLIE Monte-Carlo control converges to the optimal action-value function