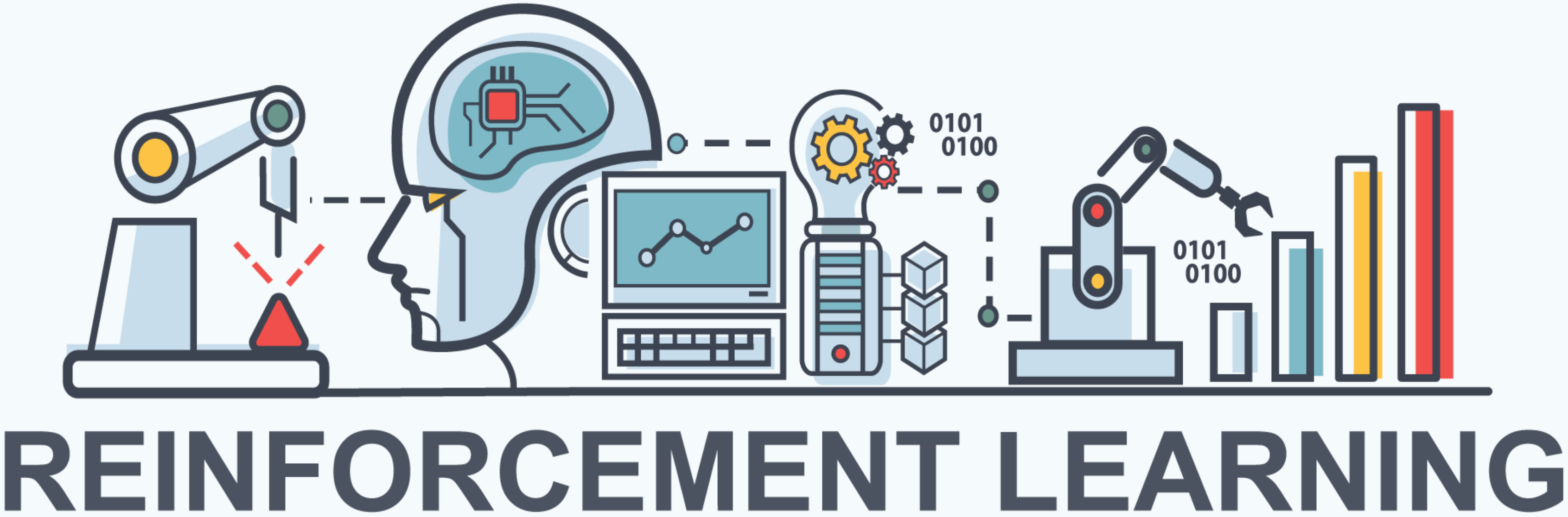


CP8319/CPS824
Lecture 9

Instructor: Nariman Farsad

* Some of the slides in this deck are adopted from courses offered David Silver, Emma Brunskill, and Sergey Levine.

Today’s Agenda

1. **Review of Previous Lectures**

2. Model-Free Control (Monte Carlo)

Markov Decision Process (MDP)

A Markov decision process (MDP) is a Markov reward process with decisions/actions

Definition

A *Markov Decision Process* is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathbf{P}, R, \gamma \rangle$

- $\mathcal{S}$ is a (finite) set of states
- $\mathcal{A}$ is a finite set of actions
- $\mathbf{P}$ is dynamics/transition model for each action,

$$P_{s,s'}^{a} = P\left(S_{t+1} = s' | S_t = s, A_t = a\right)$$

- R is the reward function, $R(s, a) = \mathbb{E}[r_t | S_t = s, A_t = a]$
- γ is a discount factor, $\gamma \in [0, 1]$

What we have learned up to now?

So far we have learned:

- Solve a *known* MDP, i.e., dynamics $\mathbf{P}$ and the reward function R are *known*
- Estimate the value function of an *unknown* MDP. i.e., dynamics $\mathbf{P}$ and the reward function R are *unknown*

Moving forward:

- Optimize the value function of an *unknown* MDP

Incremental MC (On) Policy Evaluation

Initialize $N(s) = 0, G(s) = 0 \ \forall s \in \mathcal{S}$

Loop:

- Sample episode i: $s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \ldots, s_{i,T_i}$
- Define $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \gamma^2 r_{i,t+2} + \cdots + \gamma^{T_i - 1} r_{i,T_i}$ as return from time step t onwards in i-th episode
- For each state s visited in episode i:
  - For every time t that state s is visited in episode i:
    - Increment counter of total visits: $N(s) = N(s) + 1$
    - Update estimate $v^\pi(s) = v^\pi(s) + \alpha(G_{i,t} - v^\pi(s))$

$\alpha = \frac{1}{N(s)}$: Identical to first/every visit MC

$\alpha > \frac{1}{N(s)}$: forget older data, helpful for non-stationary domains

TD(0) Policy Evaluation

- Aim: estimate $v^{\pi}(s)$ given episodes generated under policy π
- $G_t = rt + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots$ under policy π
- $v^{\pi}(s) = \mathbb{E}_{\pi}[G_t \mid S_t = s]$
- Simplest temporal-difference learning algorithm: TD(0)
  - Update value $v^{\pi}(s_t)$ toward estimated return $r_t + \gamma v^{\pi}(s_{t+1})$

$$v^{\pi}(s_t) = v^{\pi}(s_t) + \alpha([r_t + \gamma v^{\pi}(s_{t+1})] - v^{\pi}(s_t))$$

- $r_t + \gamma v^{\pi}(s_{t+1})$ is called the *TD target*
- $\delta_t = r_t + \gamma v^{\pi}(s_{t+1}) - v^{\pi}(s_t)$ is called the *TD error*
- Can immediately update value estimate after (s, a, r, s') tuple
- Don't need episodic setting

TD vs MC

- TD can learn *before* knowing the final outcome
  - TD can learn online after every step
  - MC must wait until end of episode before return is known
  - TD can learn from incomplete sequences
  - MC can only learn from complete sequences
  - TD works in continuing (non-terminating) environments
  - MC only works for episodic (terminating) environments

TD vs MC: Bias and Variance

- MC has high variance, zero bias (first-visit)
  - Good convergence properties (even with function approximation)
    - *Function approximation*: used in infinite state MDPs. We will learn about it later
  - Not very sensitive to initial values used in the initialization
  - Very simple to understand and use
- TD has low variance, some bias
  - Usually more efficient than MC
  - TD(0) converges (but not always with function approximation)
  - More sensitive to initial values used in the initialization

Today’s Agenda

1. Review of Previous Lectures

2. **Model-Free Control (Monte Carlo)**

Where is model-free control used?

Many applications can be modeled as MDPs:

- Backgammon
- Go
- Robot locomotion
- Helicopter flight
- Robocup soccer
- Autonomous driving
- Customer ad selection
- Invasive species management
- Patient treatment
- Ship steering
- Airplane logistics
- Portfolio management
- Protein folding
- Elevator
- Store inventory management
- Video games

For many of these and other problems either:

- MDP model is unknown but can be sampled
- MDP model is known but it is computationally infeasible to use directly, except through sampling

Model-free control can solve these problems.

On-Policy and Off Policy Learning

- On-policy learning
  - "Learn on the job"
  - Learn about policy π from experience sampled from π
- Off-policy learning
  - "Look over someone's shoulder"
  - Learn about policy π from experience sampled from μ (another policy)

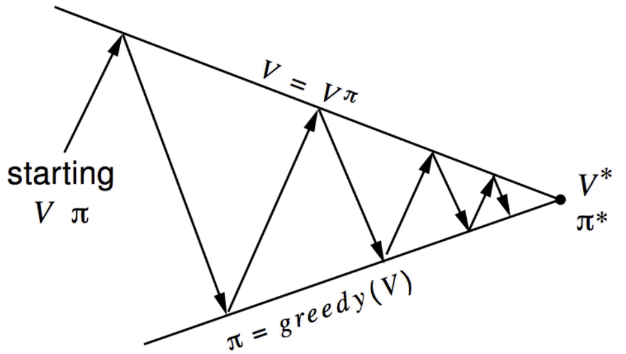
Policy Iteration: Known Model

Set $i = 0$

Initialize $\pi_0(s)$ randomly for all states s

While $i == 0$ or $\| \pi_i - \pi_{i-1} \|_1 > 0$ (L1-norm, measures if the policy changed for any state):

- $v^{\pi_i} \leftarrow$ MDP value function **policy evaluation** of π_i
- $\pi_{i+1} \leftarrow$ **Policy improvement** on v^{π_i}
- $i = i + 1$



Policy evaluation Estimate v_π
Iterative policy evaluation

Policy improvement Generate $\pi' \geq \pi$
Greedy policy improvement

Policy Evaluation: Known Model

Initialize $v_0(s) = 0$ for all s

For $k = 1$ until convergence:

For all $s \in \mathcal{S}$:

$$v_{k+1}^{\pi}(s) = \sum_{a \in \mathcal{A}} \pi(a|s)\left(R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P_{s,s'}^{a} v_k^{\pi}(s')\right) \longrightarrow$$

Action Chosen Randomly, e.g.:

- Flip a coin
- Go right if head
- Go left if tail

$$v_{k+1}^{\pi}(s) = R(s,\pi(s)) + \gamma \sum_{s' \in \mathcal{S}} P_{s,s'}^{\pi(s)} v_k^{\pi}(s') \longrightarrow$$

One action is performed deterministically:

- Always go left

This is known as Bellman expectation backup

Policy Improvement: Known Model

Compute state-action value of a policy π_i
For $s \in \mathcal{S}$ and $a \in \mathcal{A}$:

- $Q^{\pi_i}(s,a) = R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P_{s,s'}^{a} v^{\pi_i}(s')$

Compute new policy π_{i+1}, for all $s \in \mathcal{S}$

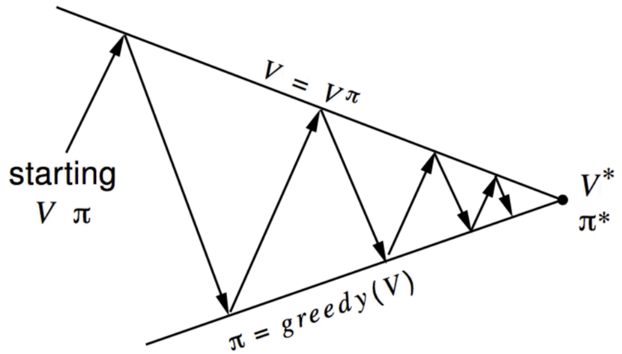
- $\pi_{i+1}(s) = \arg\max_{a \in \mathcal{A}} Q^{\pi_i}(s,a)$

With probability 1 choose an action that maximizes Q (i.e., a deterministic policy)
Hence greedy update

How can we extend to unknown models?

Set $i = 0$
Initialize $\pi_0(s)$ randomly for all states s
While $i == 0$ or $\| \pi_i - \pi_{i-1} \|_1 > 0$ (L1-norm, measures if the policy changed for any state):

- $v^{\pi_i} \leftarrow$ MDP value function **policy evaluation** of π_i
- $\pi_{i+1} \leftarrow$ **Policy improvement** on v^{π_i}
- $i = i + 1$



Policy evaluation Estimate v_π
Iterative policy evaluation

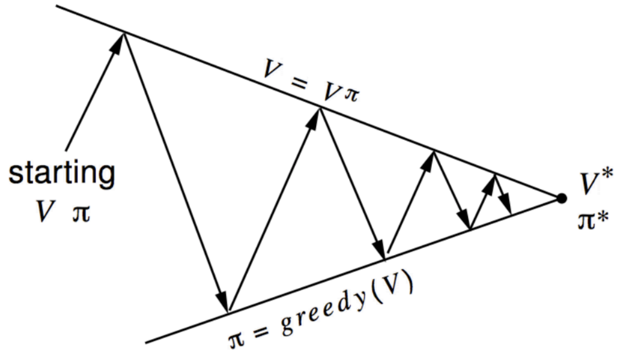
Policy improvement Generate $\pi' \geq \pi$
Greedy policy improvement

Last lecture we have learned about model-free policy evaluation. Any ideas?

Policy Iteration with MC Policy Evaluations

Set $i = 0$
Initialize $\pi_0(s)$ randomly for all states s
While $i == 0$ or $\| \pi_i - \pi_{i-1} \|_1 > 0$ (L1-norm, measures if the policy changed for any state):

- $v^{\pi_i} \leftarrow$ MDP value function **MC policy evaluation** of π_i
- $\pi_{i+1} \leftarrow$ **Policy improvement** on v^{π_i}
- $i = i + 1$



Policy evaluation Monte-Carlo policy evaluation, $V = v_\pi$?
Policy improvement Greedy policy improvement?

Do you see any problems with this?

Policy Improvement

Compute state-action value of a policy π_i
For $s \in \mathcal{S}$ and $a \in \mathcal{A}$:

- $Q^{\pi_i}(s,a) = \boxed{R(s,a)} + \gamma \sum_{s' \in \mathcal{S}} \boxed{P^a_{s,s'}} v^{\pi_i}(s')$

Calculating Q from v requires the model to be known

Compute new policy π_{i+1}, for all $s \in \mathcal{S}$

- $\pi_{i+1}(s) = \arg\max_{a \in \mathcal{A}} Q^{\pi_i}(s,a)$

How can we fix this?

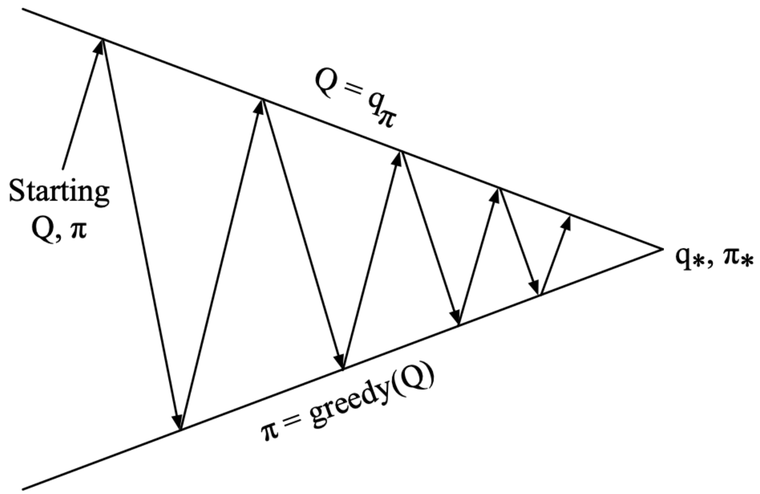
Model Free Policy Iteration

Set $i = 0$

Initialize $\pi_0(s)$ randomly for all states s

While $i == 0$ or $\| \pi_i - \pi_{i-1} \|_1 > 0$ (L1-norm, measures if the policy changed for any state):

- $Q^{\pi_i} \leftarrow$ MDP value function **MC policy Q evaluation** of π_i
- $\pi_{i+1} \leftarrow$ **Policy improvement** on Q^{π_i}
- $i = i + 1$



Policy evaluation Monte-Carlo policy evaluation, $Q = q_\pi$

Policy improvement Greedy policy improvement?

MC for On Policy Q Evaluation

Initialize $N(s,a) = 0, G(s,a) = 0, Q^{\pi}(s,a)\ \forall s \in \mathcal{S}, \forall a \in \mathcal{A}$

Loop:

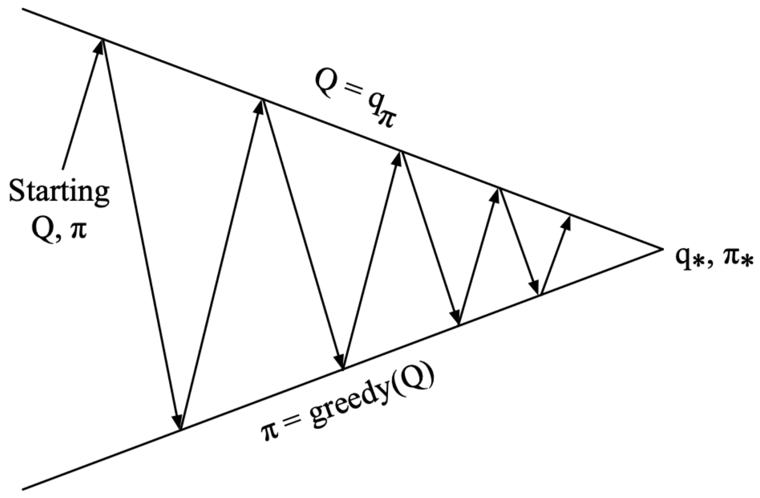
- Using policy π Sample episode i: $s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \dots, s_{i,T_i}$
- Define $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \gamma^2 r_{i,t+2} + \cdots + \gamma^{T-1} r_{i,T_i}$ as return from time step t onwards in i-th episode
- For each **state, action pair** (s,a) visited in episode i:
  - For first or every time t that state (s,a) is visited in episode i:
    - Increment counter of total visits: $N(s,a) = N(s,a) + 1$
    - Update estimate $Q^{\pi}(s,a) = Q^{\pi}(s,a) + \frac{1}{N(s,a)}(G_{i,t} - Q^{\pi}(s,a))$

Model Free Policy Iteration

Set $i = 0$
Initialize $\pi_0(s)$ randomly for all states s
While $i \ == \ 0$ or $\| \pi_i \ - \ \pi_{i-1} \|_1 > \ 0$ (L1-norm, measures if the policy changed for any state):

- $Q^{\pi_i} \leftarrow$ MDP value function **MC policy $\boldsymbol{Q}$ evaluation** of π_i
- $\pi_{i+1} \leftarrow$ **Policy improvement** on Q^{π_i}
- $i = i + 1$

Would this work? What happens when π is deterministic?

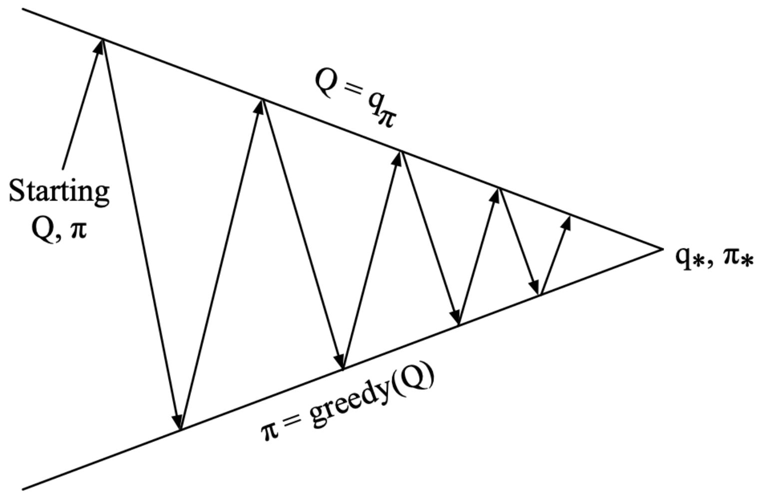


Policy evaluation Monte-Carlo policy evaluation, $Q = q_\pi$
Policy improvement Greedy policy improvement?

Model Free Policy Iteration

Set $i = 0$
Initialize $\pi_0(s)$ randomly for all states s
While $i\ == \ 0$ or $\| \pi_i - \pi_{i-1} \|_1 > \ 0$ (L1-norm, measures if the policy changed for any state):

- $Q^{\pi_i} \leftarrow$ MDP value function **MC policy *Q* evaluation** of π_i
- $\pi_{i+1} \leftarrow$ **Policy improvement** on Q^{π_i}
- $i = i + 1$



Policy evaluation Monte-Carlo policy evaluation, $Q = q_\pi$
Policy improvement Greedy policy improvement?

Would this work? What happens when π is deterministic?

- If π is deterministic, policy Q evaluation can't compute $Q(s, a)$ for any $a\ \neq \pi(s)$
- May not converge to optimal policy

Policy Evaluation with Exploration

- Want to compute a model-free estimate of Q^{π}
- In general, seems subtle
  - Need to try all (s, a) pairs but then follow π
  - Want to ensure resulting estimate Q^{π} is good enough so that policy improvement is a monotonic operator
- For certain classes of policies can ensure all (s, a) pairs are tried such that asymptotically Q^{π} converges to the true value

ϵ-greedy Policies

- Simple idea to balance exploration and exploitation
- Let $m = |\mathcal{A}|$ be the number of actions
- Then an ϵ-greedy policy w.r.t. a state-action value $Q(s, a)$ is

$$\boxed{\pi(a|s) = \begin{cases} \epsilon/m + 1 - \epsilon\,, & \text{if } a = \operatorname*{argmax}_{a' \in \mathcal{A}} Q(s, a) \\ \epsilon/m \quad\quad\quad\quad\,, & otherwise \end{cases}}$$

- With probability 1 – ϵ choose the greedy action
- With probability ϵ choose an action at random

Model Free Policy Iteration

Set $i = 0$

Initialize $\pi_0(s)$ randomly for all states s

While $i == 0$ or $\| \pi_i - \pi_{i-1} \|_1 > 0$ (L1-norm, measures if the policy changed for any state):

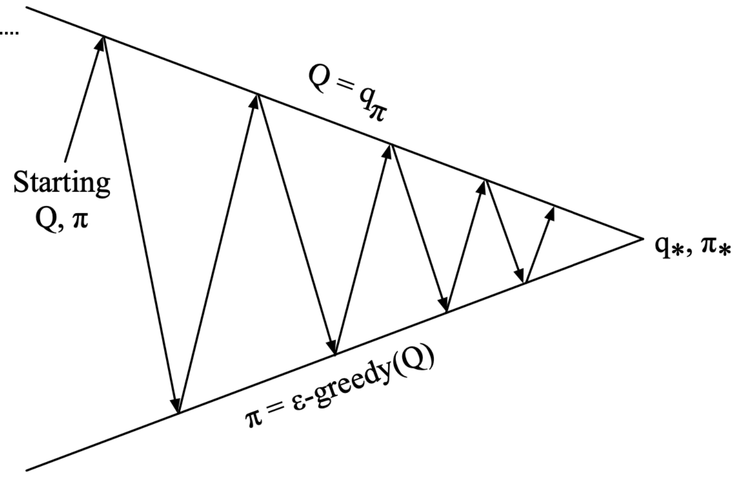
- $Q^{\pi_i} \leftarrow$ MDP value function **MC policy Q evaluation** of π_i
- $\pi_{i+1} \leftarrow$ **ϵ-greedy Policy improvement** on Q^{π_i}
- $i = i + 1$

greedy(Q)

$$\pi_{i+1}(a|s) = \begin{cases} 1, & \text{if } a = \underset{a' \in \mathcal{A}}{\operatorname{argmax}}\, Q^{\pi_i}(s,a') \\ 0, & otherwise \end{cases}$$

ϵ-greedy(Q)

$$\pi_{i+1}(a|s) = \begin{cases} \epsilon/m + 1 - \epsilon\,, & \text{if } a = \underset{a' \in \mathcal{A}}{\operatorname{argmax}}\, Q^{\pi_i}(s,a') \\ \epsilon/m \qquad\qquad , & otherwise \end{cases}$$



Policy evaluation Monte-Carlo policy evaluation, $Q = q_\pi$

Policy improvement ϵ-greedy policy improvement

ϵ-Greedy MC for On Policy Q Evaluation

s_1	s_2	s_3	s_4	s_5	s_6	s_7
						

- Robot with two actions
  - R(-, a_1) = [1 0 0 0 0 0 +10] and R(-, a_2) = [0 0 0 0 0 0 +5]
- $\pi(s) = a_1 \ \forall s, \gamma = 1, \epsilon = 0.5$. Any action from s1 and s7 terminates episode
- Sample episode = $(s_3, a_1, 0, s_2, a_2, 0, s_3, a_1, 0, s_2, a_2, 0, s_1, a_1, 1$ terminal)
- First visit MC estimate of Q of each (s, a) pair?

Monotonic ϵ-Greedy Policy Improvement

Theorem

For any ϵ-greedy policy π_i, the ϵ-greedy policy w.r.t. Q^{π_i}, π_{i+1} is a monotonic improvement $V^{\pi_{i+1}} \geq V^{\pi_i}$

$$
\begin{aligned}
Q^{\pi_i}(s, \pi_{i+1}(s)) &= \sum_{a \in A} \pi_{i+1}(a|s) Q^{\pi_i}(s, a) \\
&= (\epsilon/|A|) \left[\sum_{a \in A} Q^{\pi_i}(s, a)\right] + (1-\epsilon) \max_a Q^{\pi_i}(s, a) \\
&= (\epsilon/|A|) \left[\sum_{a \in A} Q^{\pi_i}(s, a)\right] + (1-\epsilon) \max_a Q^{\pi_i}(s, a) \frac{1-\epsilon}{1-\epsilon} \\
&= (\epsilon/|A|) \left[\sum_{a \in A} Q^{\pi_i}(s, a)\right] + (1-\epsilon) \max_a Q^{\pi_i}(s, a) \sum_{a \in A} \frac{\pi_i(a|s) - \frac{\epsilon}{|A|}}{1-\epsilon} \\
&\geq \frac{\epsilon}{|A|} \left[\sum_{a \in A} Q^{\pi_i}(s, a)\right] + (1-\epsilon) \sum_{a \in A} \frac{\pi_i(a|s) - \frac{\epsilon}{|A|}}{1-\epsilon} Q^{\pi_i}(s, a) \\
&= \sum_{a \in A} \pi_i(a|s) Q^{\pi_i}(s, a) = V^{\pi_i}(s)
\end{aligned}
$$

Each step of policy improvement improves policy or keeps it the same

Greedy in the Limit of Infinite Exploration (GLIE)

Definition of GLIE

- All state-action pairs are visited an infinite number of times

$$\lim_{i \to \infty} N_i(s, a) \to \infty$$

- Behavior policy (policy used to act in the world) converges to greedy policy
  $\lim_{i \to \infty} \pi(a|s) \to \arg\max_a Q(s, a)$ with probability 1

- A simple GLIE strategy is ϵ-greedy where ϵ is reduced to 0 with the following rate: $\epsilon_i = 1/i$

Monte Carlo Online Control/On Policy Improvement

Initialize $Q(s,a) = 0, N(s,a) = 0\ \forall(s,a)$, Set $\epsilon = 1,\ k = 1$
$\pi_k = \epsilon$-greedy(Q) // Create initial ϵ-greedy policy
loop
    Sample k-th episode $(s_{k,1}, a_{k,1}, r_{k,1}, s_{k,2}, \ldots, s_{k,T})$ given π_k
    $G_{k,t} = r_{k,t} + \gamma r_{k,t+1} + \gamma^2 r_{k,t+2} + \cdots \gamma^{T_i - 1} r_{k,T_i}$
    for $t = 1, \ldots, T$ **do**
        if First visit to (s,a) in episode k **then**
            $N(s,a) = N(s,a) + 1$
            $Q(s_t, a_t) = Q(s_t, a_t) + \frac{1}{N(s,a)}(G_{k,t} - Q(s_t, a_t))$
        end if
    end for
    $k = k + 1,\ \epsilon = 1/k$
    $\pi_k = \epsilon$-greedy(Q) // Policy improvement
end loop

MC for On Policy Control Example

s_1	s_2	s_3	s_4	s_5	s_6	s_7
						

- Robot with two actions
  - R(-, a_1) = [1 0 0 0 0 0 +10] and R(-, a_2) = [0 0 0 0 0 0 +5]
- $\pi(s) \ = \ a_1 \ \forall s, \gamma = 1, \epsilon = 0.5$. Any action from s1 and s7 terminates episode
- Sample episode = $(s_3, a_1, 0, s_2, a_2, 0, s_3, a_1, 0, s_2, a_2, 0, s_1, a_1, 1$ terminal)
- First visit MC estimate of Q of each (s, a) pair?
  - $Q^\pi(-, a_1) = [1\ 0\ 1\ 0\ 0\ 0\ 0], \ \ Q^\pi(-, a_2) = [0\ 1\ 0\ 0\ 0\ 0\ 0]$
- What is $\pi(s) = \arg\max_a Q^\pi(s, a) \ \forall \mathcal{S}$?

- What is new ϵ-greedy policy, if $k \ = \ 3, \ \ \epsilon \ = \ 1/k$? Give an example for $\pi(s_1)$.