# Machine Learning I
# MATH60629A

Sequential Decision Making II
**Summary**
— Week #13

# Brief recap

- Markov Decision Processes (MDP)
  - Offer a framework for sequential decision making

    $$\langle \mathsf{A}, \mathsf{S}, \mathsf{P}, \mathsf{R}, \gamma \rangle$$

  - Goal: find the optimal policy
    - Dynamic programming and several algorithms (e.g., VI,PI)

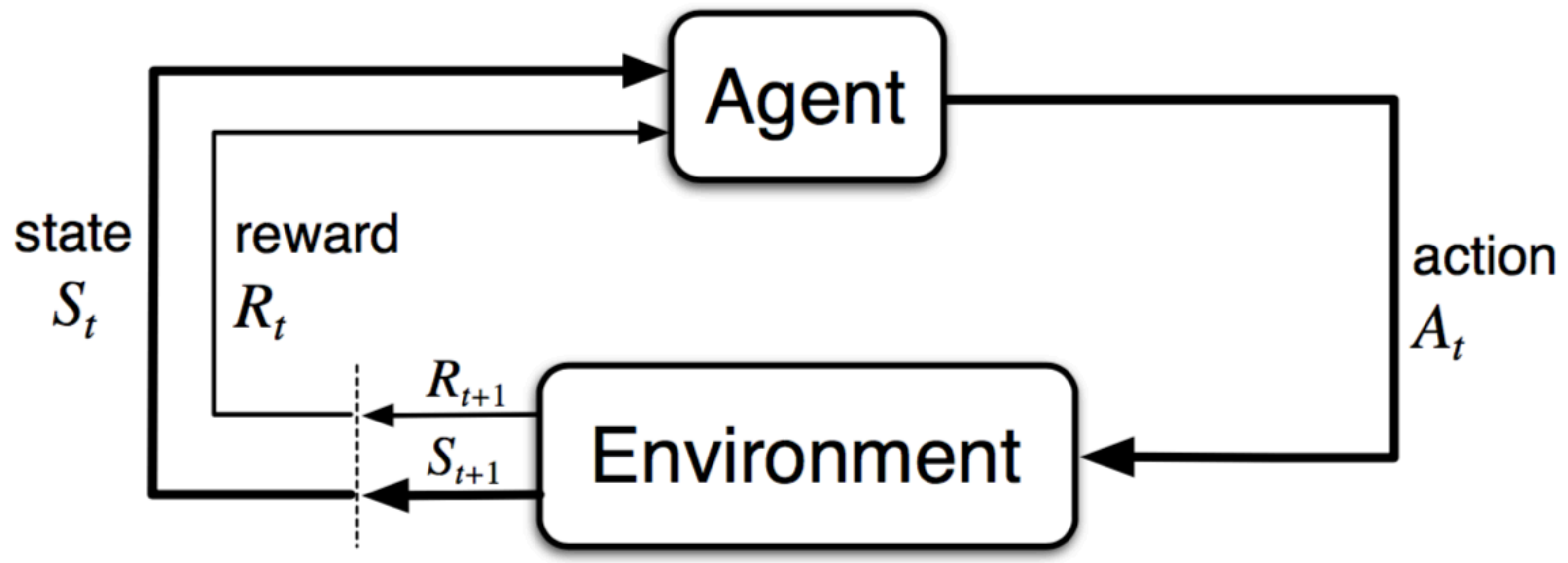


Figure 3.1, RL: An introduction

# From MDPs to RL

- In MDPs we assume that we know

# From MDPs to RL

- In MDPs we assume that we know
  1. Transition probabilities: $P(s' \mid s, a)$

# From MDPs to RL

- In MDPs we assume that we know

  1. Transition probabilities: $P(s' \mid s, a)$

  2. Reward function: $R(s)$

# From MDPs to RL

- In MDPs we assume that we know
  1. Transition probabilities: $P(s' \mid s, a)$
  2. Reward function: $R(s)$
- RL is more general

# From MDPs to RL

- In MDPs we assume that we know
  1. Transition probabilities: $P(s' \mid s, a)$
  2. Reward function: $R(s)$
- RL is more general
  - In RL both are typically unknown

# From MDPs to RL

- In MDPs we assume that we know
  1. Transition probabilities: P(s' | s, a)
  2. Reward function: R(s)
- RL is more general
  - In RL both are typically unknown
  - RL agents navigate the world to gather this information

# Experience

A. Supervised Learning:
   - Given fixed dataset
   - Goal: maximize objective on test set (population)

B. Reinforcement Learning
   - Collect data as agent interacts with the world
   - Goal: maximize sum of rewards

# Algorithms for Reinforcement Learning

# Monte Carlo Methods

# Monte Carlo Methods

- **Model-free**

# Monte Carlo Methods

- Model-free
- Assume the environment is episodic
  - Think of playing a card game (like poker). An episode is a hand.
  - Updates the policy after each episode

# Monte Carlo Methods

- Model-free
- Assume the environment is episodic
  - Think of playing a card game (like poker). An episode is a hand.
  - Updates the policy after each episode
- Intuition
  - Experience many episodes
    - Play many hands (of poker)
  - Average the rewards received at each state
    - What is the proportion of wins given your curent cards

# First-visit Monte Carlo

- Given a fixed policy (prediction)

- Calculate the value function V(s) for each state

**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 $s \in \mathcal{S}$

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 $Returns(s)$
        $V(s) \leftarrow \text{average}(Returns(s))$

[Sutton & Barto, RL Book, Ch 5]

- Converges to $V_\pi(s)$ as the number of visits to each state goes to infinity

# First-visit Monte Carlo

- Given a fixed policy (prediction)
- Calculate the value function V(s) for each state

**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 $s \in \mathcal{S}$

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

[Sutton & Barto, RL Book, Ch 5]

- Converges to $V_\pi(s)$ as the number of visits to each state goes to infinity

$$\boxed{V(s_t) = \max_{a_t} \left\{ R(s_t) + \gamma \sum_{s_{t+1}} P(s_{t+1} \mid s_t, a_t) V(s_{t+1}) \right\}}$$

# Example: Black Jack

- Episode: one hand
- States: Sum of player’s cards, dealer’s card, usable ace
- Actions: {Stay, Hit}
- Rewards: {Win +1, Tie 0, Loose -1}
- A few other assumptions: infinite deck

# Q-value function for control

- We know about state-value functions $V(s)$

# Q-value function for control

- We know about state-value functions V(s)
  - If state transitions are known then they can be used to derive an optimal policy [recall value iteration]:

$$\boldsymbol{\pi}^*(\mathbf{s}) = \arg\max_{\mathbf{a}} \left\{ \mathbf{R}(\mathbf{s}) + \gamma \sum_{\mathbf{s}'} \mathbf{P}(\mathbf{s}' \mid \mathbf{s}, \mathbf{a}) \mathbf{V}^*(\mathbf{s}') \right\} \; \forall \mathbf{s}$$

# Q-value function for control

- We know about state-value functions V(s)

  - If state transitions are known then they can be used to derive an optimal policy [recall value iteration]:

$$\boldsymbol{\pi}^*(\mathbf{s}) = \arg\max_{\mathbf{a}} \left\{ \mathbf{R}(\mathbf{s}) + \gamma \sum_{\mathbf{s}'} \mathbf{P}(\mathbf{s}' \mid \mathbf{s}, \mathbf{a}) \mathbf{V}^*(\mathbf{s}') \right\} \ \forall \mathbf{s}$$

  - When state transitions are unknown what can we do?

# Q-value function for control

- We know about state-value functions V(s)
  - If state transitions are known then they can be used to derive an optimal policy [recall value iteration]:

$$\boldsymbol{\pi}^*(\mathbf{s}) = \arg\max_{\mathbf{a}} \left\{ \mathbf{R}(\mathbf{s}) + \gamma \sum_{\mathbf{s}'} \mathbf{P}(\mathbf{s}' \mid \mathbf{s}, \mathbf{a}) \mathbf{V}^*(\mathbf{s}') \right\} \ \forall \mathbf{s}$$

  - When state transitions are unknown what can we do?
    - Q(s,a) the value function of a (state,action) pair

$$\boldsymbol{\pi}^*(\mathbf{s}) = \arg\max_{\mathbf{a}} \{ \mathbf{Q}^*(\mathbf{s}, \mathbf{a}) \} \ \forall \mathbf{s}$$

# Monte Carlo without exploring starts (on policy)

**On-policy first-visit MC control (for $\varepsilon$-soft policies), estimates $\pi \approx \pi_*$**

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

Repeat forever:
  (a) Generate an episode using $\pi$
  (b) For each pair $s, a$ appearing in the episode:
    $G \leftarrow$ the return that follows the first occurrence of $s, a$
    Append $G$ to $Returns(s,a)$
    $Q(s,a) \leftarrow \text{average}(Returns(s,a))$
  (c) For each $s$ in the episode:
    $A^* \leftarrow \arg\max_a Q(s,a)$ (with ties broken arbitrarily)
    For all $a \in \mathcal{A}(s)$:

$$\pi(a|s) \leftarrow \begin{cases} 1 - \varepsilon + \varepsilon/|\mathcal{A}(s)| & \text{if } a = A^* \\ \varepsilon/|\mathcal{A}(s)| & \text{if } a \neq A^* \end{cases}$$

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

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 $\pi$
  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:
    $\pi(s) \leftarrow \arg\max_a Q(s,a)$

[Sutton & Barto, RL Book, Ch.5]

# Monte Carlo without exploring starts (on policy)

**On-policy first-visit MC control (for $\varepsilon$-soft policies), estimates $\pi \approx \pi_*$**

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
- $Q(s, a) \leftarrow$ arbitrary
- $Returns(s, a) \leftarrow$ empty list
- $\boxed{\pi(a|s) \leftarrow \text{an arbitrary } \varepsilon\text{-soft policy}}$

Repeat forever:
- (a) Generate an episode using $\pi$
- (b) For each pair $s, a$ appearing in the episode:
  - $G \leftarrow$ the return that follows the first occurrence of $s, a$
  - Append $G$ to $Returns(s, a)$
  - $Q(s, a) \leftarrow \text{average}(Returns(s, a))$
- (c) For each $s$ in the episode:
  - $A^* \leftarrow \arg\max_a Q(s, a)$ (with ties broken arbitrarily)
  - For all $a \in \mathcal{A}(s)$:

$$\pi(a|s) \leftarrow \begin{cases} 1 - \varepsilon + \varepsilon/|\mathcal{A}(s)| & \text{if } a = A^* \\ \varepsilon/|\mathcal{A}(s)| & \text{if } a \neq A^* \end{cases}$$

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

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 $\pi$
- 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:
  - $\pi(s) \leftarrow \arg\max_a Q(s, a)$

[Sutton & Barto, RL Book, Ch.5]

# Monte Carlo without exploring starts (on policy)

**On-policy first-visit MC control (for $\varepsilon$-soft policies), estimates $\pi \approx \pi_*$**

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
    $Q(s,a) \leftarrow$ arbitrary
    $Returns(s,a) \leftarrow$ empty list
    $\boxed{\pi(a|s) \leftarrow \text{an arbitrary } \varepsilon\text{-soft policy}}$

Repeat forever:
    (a) Generate an episode using $\pi$
    (b) For each pair $s, a$ appearing in the episode:
        $G \leftarrow$ the return that follows the first occurrence of $s, a$
        Append $G$ to $Returns(s,a)$
        $Q(s,a) \leftarrow \text{average}(Returns(s,a))$
    (c) For each $s$ in the episode:
        $A^* \leftarrow \arg\max_a Q(s,a)$    (with ties broken arbitrarily)
        For all $a \in \mathcal{A}(s)$:

$$\boxed{\pi(a|s) \leftarrow \begin{cases} 1 - \varepsilon + \varepsilon/|\mathcal{A}(s)| & \text{if } a = A^* \\ \varepsilon/|\mathcal{A}(s)| & \text{if } a \neq A^* \end{cases}}$$

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

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 $\pi$
    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:
        $\pi(s) \leftarrow \arg\max_a Q(s,a)$

[Sutton & Barto, RL Book, Ch.5]

# Monte Carlo without exploring starts (on policy)

**On-policy first-visit MC control (for $\varepsilon$-soft policies), estimates $\pi \approx \pi_*$**

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
  $Q(s,a) \leftarrow$ arbitrary
  $Returns(s,a) \leftarrow$ empty list
  $\boxed{\pi(a|s) \leftarrow \text{an arbitrary } \varepsilon\text{-soft policy}}$

Repeat forever:
  (a) Generate an episode using $\pi$
  (b) For each pair $s, a$ appearing in the episode:
    $G \leftarrow$ the return that follows the first occurrence of $s, a$
    Append $G$ to $Returns(s,a)$
    $Q(s,a) \leftarrow \text{average}(Returns(s,a))$
  (c) For each $s$ in the episode:
    $A^* \leftarrow \arg\max_a Q(s,a)$     (with ties broken arbitrarily)
    For all $a \in \mathcal{A}(s)$:

$$\boxed{\pi(a|s) \leftarrow \begin{cases} 1 - \varepsilon + \varepsilon/|\mathcal{A}(s)| & \text{if } a = A^* \\ \varepsilon/|\mathcal{A}(s)| & \text{if } a \neq A^* \end{cases}}$$

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

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 $\pi$
  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:
    $\pi(s) \leftarrow \arg\max_a Q(s,a)$

[Sutton & Barto, RL Book, Ch.5]

- Policy value cannot decrease

$$v_{\pi'}(s) \geq v_\pi(s), \forall s \in S$$

$\pi$ : policy at current step
$\pi'$ : policy at next step

# Monte-Carlo methods summary

- Allow a policy to be learned through interactions
  - (Does not learn transitions)
- States are effectively treated as being independent
  - Focus on a subset of states (e.g., states for which playing optimally is of particular importance)
- Episodic (with or without exploring starts)

# TD for control

**Sarsa (on-policy TD control) for estimating $Q \approx q_*$**

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(\mathit{terminal}, \cdot) = 0$

Loop for each episode:
    Initialize $S$
    Choose $A$ from $S$ using policy derived from $Q$ (e.g., $\varepsilon$-greedy)
    Loop for each step of episode:
        Take action $A$, observe $R$, $S'$
        Choose $A'$ from $S'$ using policy derived from $Q$ (e.g., $\varepsilon$-greedy)
        $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma Q(S', A') - Q(S, A)\big]$
        $S \leftarrow S'$; $A \leftarrow A'$;
    until $S$ is terminal

**Tabular TD(0) for estimating $v_\pi$**

Input: the policy $\pi$ to be evaluated
Initialize $V(s)$ arbitrarily (e.g., $V(s) = 0$, for all $s \in \mathcal{S}^+$)
Repeat (for each episode):
    Initialize $S$
    Repeat (for each step of episode):
        $A \leftarrow$ action given by $\pi$ for $S$
        Take action $A$, observe $R$, $S'$
        $V(S) \leftarrow V(S) + \alpha\big[R + \gamma V(S') - V(S)\big]$
        $S \leftarrow S'$
    until $S$ is terminal

# Comparing TD and MC

- MC requires going through full episodes before updating the value function. Episodic.
- Converges to the optimal solution

- TD updates each V(s) after each transition. Online.
- Converges to the optimal solution (some conditions on $\alpha$)
- Empirically TD methods tend to converge faster