Markov Decision Process

Prabuchandran K.J.

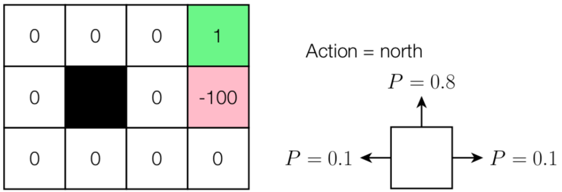
Assistant Professor, IIT Dharwad

9 March 2020

Outline

- Grid World
- Markov Decision Process (MDP)
- Bellman Equation
- Solution Methodolgy

Example Simple MDP: Gridworld



Grid World with discount factor $\gamma = 0.9$

Grid World Setup

- Simple grid world with a ‘goal state’ with reward and a ‘bad state’ with reward -100
- Actions move in the desired direction with probabilty 0.8
- Taking an action that would bump into a wall leaves agent where it is

Grid World MDP

0	0	0	1
0		0	-100
0	0	0	0

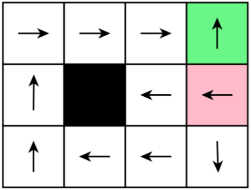
Reward Function

Grid World MDP

5.470	6.313	7.190	8.669
4.802		3.347	-96.67
4.161	3.654	3.222	1.526

Value Function

Grid World MDP



Optimal Policy

A Finite Markov Decision Process (MDP)

- State Space, $S = \{1, 2, \ldots, N\}$

A Finite Markov Decision Process (MDP)

- State Space, $S = \{1, 2, \ldots, N\}$
- Action Space, $A = \{1, 2, \ldots, M\}$

A Finite Markov Decision Process (MDP)

- State Space, $S = \{1, 2, \ldots, N\}$

- Action Space, $A = \{1, 2, \ldots, M\}$

- Probability transition kernel, $P_{i,j}(a)$

$$Pr\{s_{n+1} = j | s_n = i, a_n = a\} = P_{i,j}(a)$$

s_n, s_{n+1} - state at time n and $n+1$, a_n - action at time n.

A Finite Markov Decision Process (MDP)

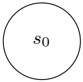
- State Space, $S = \{1, 2, \ldots, N\}$
- Action Space, $A = \{1, 2, \ldots, M\}$
- Probability transition kernel, $P_{i,j}(a)$

$$Pr\{s_{n+1} = j | s_n = i, a_n = a\} = P_{i,j}(a)$$

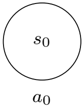
s_n, s_{n+1} - state at time n and $n+1$, a_n - action at time n.

- Reward function, $R(i, a, j)$

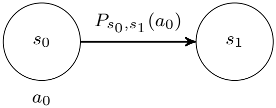
Markov Decision Process (MDP) Evolution



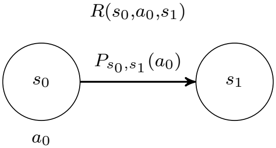
Markov Decision Process (MDP) Evolution



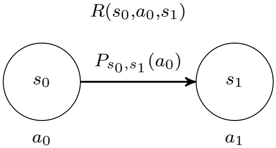
Markov Decision Process (MDP) Evolution



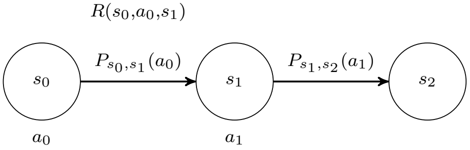
Markov Decision Process (MDP) Evolution



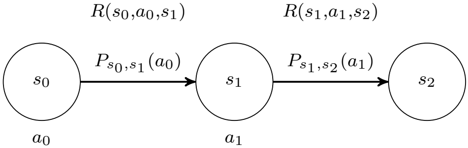
Markov Decision Process (MDP) Evolution



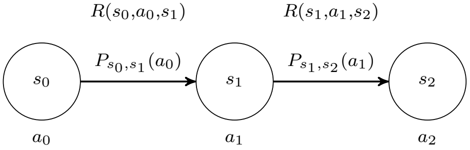
Markov Decision Process (MDP) Evolution



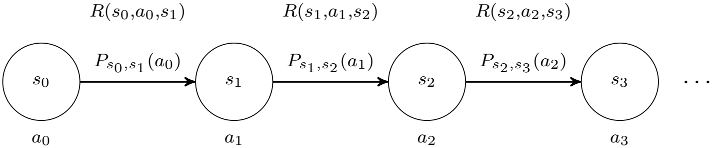
Markov Decision Process (MDP) Evolution



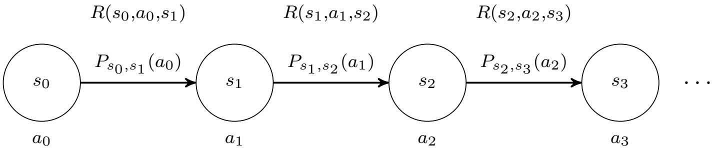
Markov Decision Process (MDP) Evolution



Markov Decision Process (MDP) Evolution



Markov Decision Process (MDP) Evolution



Goal : Find the optimal sequence of actions to maximize a long-term objective

Long-term objective

Total reward obtained in an infinite horizon

$$\mathbb{E}\left[\underbrace{R(s_0, a_0, s_1)}_{R_1} + \underbrace{R(s_1, a_1, s_2)}_{R_2} + \underbrace{R(s_2, a_2, s_3)}_{R_3} + \cdots\right]$$

Long-term objective

Total reward obtained in an infinite horizon

$$\mathbb{E}\left[\underbrace{R(s_0, a_0, s_1)}_{R_1} + \underbrace{R(s_1, a_1, s_2)}_{R_2} + \underbrace{R(s_2, a_2, s_3)}_{R_3} + \cdots\right]$$

Goal: Find $\{a_0^*, a_1^*, a_2^*, \cdots\}$ that maximize total reward

Long-term objective

Discounted reward obtained in an infinite horizon

$$\mathbb{E}\left[\underbrace{R(s_0, a_0, s_1)}_{R_1} + \gamma \underbrace{R(s_1, a_1, s_2)}_{R_2} + \gamma^2 \underbrace{R(s_2, a_2, s_3)}_{R_3} + \cdots\right]$$

Long-term objective

Discounted reward obtained in an infinite horizon

$$\mathbb{E}\left[\underbrace{R(s_0, a_0, s_1)}_{R_1} + \gamma\, \underbrace{R(s_1, a_1, s_2)}_{R_2} + \gamma^2\, \underbrace{R(s_2, a_2, s_3)}_{R_3} + \cdots\right]$$

Goal: Find $\{a_0^*, a_1^*, a_2^*, \cdots\}$ that maximize long-run discounted reward

Long-term objective

Average reward obtained in an infinite horizon

$$\lim_{T\to\infty} \mathbb{E}\left[\frac{\overbrace{R(s_0, a_0, s_1)}^{R_1} + \overbrace{R(s_1, a_1, s_2)}^{R_2} + \cdots + \overbrace{R(s_{T-1}, a_{T-1}, s_T)}^{R_T}}{T}\right]$$

Long-term objective

Average reward obtained in an infinite horizon

$$\lim_{T\to\infty} \mathbb{E}\left[\frac{\overbrace{R(s_0, a_0, s_1)}^{R_1} + \overbrace{R(s_1, a_1, s_2)}^{R_2} + \cdots + \overbrace{R(s_{T-1}, a_{T-1}, s_T)}^{R_T}}{T}\right]$$

Goal: Find $\{a_0^*, a_1^*, a_2^*, \cdots\}$ that maximize long-run average reward

Policy: State Dependent Actions

Stationary Deterministic Policy (SDP)

State	Action
1	$\mu(1)$
2	$\mu(2)$
$\vdots$	$\vdots$
N	$\mu(N)$

Policy $\mu \colon S \rightarrow A$

Long term dependencies

How good is a policy ?

Value function

Different Starting states

$$1 \rightarrow \mathbb{E}[R(1, \mu(1), s_1) + \gamma \; R(s_1, \mu(s_1), s_2) + \gamma^2 \; R(s_2, \mu(s_2), s_3) \cdots]$$

Long term dependencies

How good is a policy ?

Value function

Different Starting states

$$i \rightarrow \mathbb{E}[R(i, \mu(i), s_1) + \gamma\ R(s_1, \mu(s_1), s_2) + \gamma^2\ R(s_2, \mu(s_2), s_3) \cdots]$$

Discounted Reward Value Function

State	Value
1	$V^{\mu}(1)$
2	$V^{\mu}(2)$
$\vdots$	$\vdots$
N	$V^{\mu}(N)$

Value function $V^{\mu}: S \to \mathbb{R}$

$$V^{\mu}(i) = \sum_{n=0}^{\infty} \mathbb{E}[\gamma^n R(s_n, a_n, s_{n+1}) | s_0 = i, \mu], \tag{1}$$

Consecutive Heads Puzzle

- Experiment - Toss fair coin until we get two consecutive heads
- Let N denote the number of tosses required to get two consecutive heads
- $\mathbb{E}[N]$ - Expected number of trials to get consecutive heads
- How to compute $\mathbb{E}[N]$?

Naive Approach

Trials	Probability
2	$\frac{1}{4}$
3	p_3
$\vdots$	$\vdots$
n	p_n
$\vdots$	$\vdots$

Probability of getting two consecutive heads in exactly n trials

$$\mathbb{E}[N] = 1 * p_1 + 2 * p_2 + 3 * p_3 + 4 * p_4 + \dots$$
$$= \sum_{i=1}^{\infty} i * p_i$$

Another Approach

- Let $x = \mathbb{E}[N]$

Another Approach

- Let $x = \mathbb{E}[N]$
- what is $\mathbb{E}[N | \text{First toss is Tail}]$?

Another Approach

- Let $x = \mathbb{E}[N]$
- what is $\mathbb{E}[N|\text{First toss is Tail}]$?
- $\mathbb{E}[N|T] = x + 1$

Another Approach

- Let $x = \mathbb{E}[N]$
- what is $\mathbb{E}[N|\text{First toss is Tail}]$?
- $\mathbb{E}[N|T] = x + 1$
- what is $\mathbb{E}[N|H]$?

Another Approach

- Let $x = \mathbb{E}[N]$
- what is $\mathbb{E}[N|\text{First toss is Tail}]$?
- $\mathbb{E}[N|T] = x + 1$
- what is $\mathbb{E}[N|H]$?
- Can't say immediately

How to Aggregate?

- Assume we know $\mathbb{E}[N|H]$

How to Aggregate?

- Assume we know $\mathbb{E}[N|H]$
- How to compute $\mathbb{E}[N]$

How to Aggregate?

- Assume we know $\mathbb{E}[N|H]$
- How to compute $\mathbb{E}[N]$
- $\mathbb{E}[N] = \frac{1}{2}\mathbb{E}[N|H] + \frac{1}{2}\mathbb{E}[N|T]$

How to compute $\mathbb{E}[N|H]$?

- What is $\mathbb{E}[N|H, H]$?

How to compute $\mathbb{E}[N|H]$?

- What is $\mathbb{E}[N|H,H]$?

- $\mathbb{E}[N|H,H] = 2$

How to compute $\mathbb{E}[N|H]$?

- What is $\mathbb{E}[N|H,H]$?
- $\mathbb{E}[N|H,H] = 2$
- What is $\mathbb{E}[N|H,T]$?

How to compute $\mathbb{E}[N|H]$?

- What is $\mathbb{E}[N|H,H]$?
- $\mathbb{E}[N|H,H] = 2$
- What is $\mathbb{E}[N|H,T]$?
- $\mathbb{E}[N|H,H] = x + 2$
- What is $\mathbb{E}[N|H,H]$?

How to compute $\mathbb{E}[N|H]$?

- What is $\mathbb{E}[N|H,H]$?
- $\mathbb{E}[N|H,H] = 2$
- What is $\mathbb{E}[N|H,T]$?
- $\mathbb{E}[N|H,H] = x + 2$
- What is $\mathbb{E}[N|H,H]$?
- $\mathbb{E}[N|H] = \frac{1}{2}\mathbb{E}[N|H,H] + \frac{1}{2}\mathbb{E}[N|H,T]$

Recursive Approach

Trials	Probability
$\mathbb{E}[N\|H,H]$	2
$\mathbb{E}[N\|H,T]$	$x+2$
$\mathbb{E}[N\|H]$	$\frac{1}{2}2+\frac{1}{2}(x+2)$
$\mathbb{E}[N\|T]$	$x+1$
$\mathbb{E}[N]$	$\frac{1}{2}\mathbb{E}[N\|H]+\frac{1}{2}\mathbb{E}[N\|T]$

Recursion Table

$$x = \frac{1}{2}(x+1) + \frac{1}{2}(\frac{1}{2}(2) + \frac{1}{2}(x+2))$$

$$= 6$$

Bellman Equation for a Fixed Policy μ

$$V^{\mu}(i) = \sum_{j=1}^{N} P_{ij}(\mu(i))[R(i,\mu(i),j) + \gamma V^{\mu}(j)] \tag{2}$$

$$V^{\mu} = R^{\mu} + \gamma P^{\mu} V^{\mu}, \tag{3}$$

P^{μ} - transition probabilities between states under μ,

R^{μ} - vector of single-stage costs

Bellman Equation for a Fixed Policy μ

$$V^{\mu}(i) = \sum_{j=1}^{N} P_{ij}(\mu(i))[R(i,\mu(i),j) + \gamma V^{\mu}(j)] \tag{2}$$

$$V^{\mu} = R^{\mu} + \gamma P^{\mu} V^{\mu}, \tag{3}$$

P^{μ} - transition probabilities between states under μ,

R^{μ} - vector of single-stage costs

Long term cost $=$ Expected immediate cost $+$ Expected future cost

Goal for Long-run Discounted Reward Objective

- Optimal value function V^*

$$V^*(i) = \min_{\mu \in \Pi} V_\mu(i), \ \forall i \in S, \tag{4}$$

Π - set of all SDPs

Goal for Long-run Discounted Reward Objective

- Bellman Equation

$$V^*(i) = \max_{a \in A} \sum_{j \in S} P_{ij}(a) \left[R(i,a,j) + \gamma V^*(j) \right] \; \forall i \in S. \tag{5}$$

- Optimal SDP μ^*

$$\mu^*(i) = \arg\min_{a \in A} \sum_{j \in S} P_{ij}(a) \left[R(i,a,j) + \gamma V^*(j) \right] \; \forall i \in S. \tag{6}$$

Steps to find Optimal Policy

Goal Find optimal sequence of actions to maximize the given objective

Steps to find Optimal Policy

Goal Find optimal sequence of actions to maximize the given objective

1. Policy representation of actions

Steps to find Optimal Policy

Goal Find optimal sequence of actions to maximize the given objective

1. Policy representation of actions
2. Associate value function V^{μ} for policy μ

Steps to find Optimal Policy

Goal Find optimal sequence of actions to maximize the given objective

1. Policy representation of actions
2. Associate value function V^{μ} for policy μ
3. Finite number of stationary deterministic policies M^N

Steps to find Optimal Policy

Goal Find optimal sequence of actions to maximize the given objective

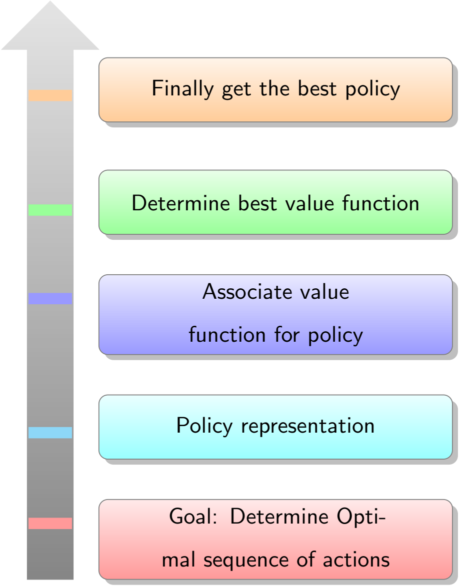
1. Policy representation of actions
2. Associate value function V^{μ} for policy μ
3. Finite number of stationary deterministic policies M^{N}
4. Find optimal value function V^{*} max of all value function vectors V^{μ}s

Steps to find Optimal Policy

Goal Find optimal sequence of actions to maximize the given objective

1. Policy representation of actions
2. Associate value function V^μ for policy μ
3. Finite number of stationary deterministic policies M^N
4. Find optimal value function V^* max of all value function vectors V^μs
5. Find optimal stationary deterministic policy μ^* from V^*

Summary


Finally get the best policy
Determine best value function
Associate value function for policy
Policy representation
Goal: Determine Optimal sequence of actions

Naive Solution

- Compute value function for all the policies

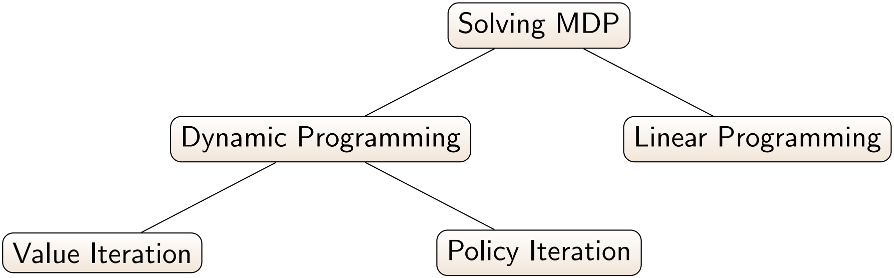
Naive Solution

- Compute value function for all the policies
- Find the best value function

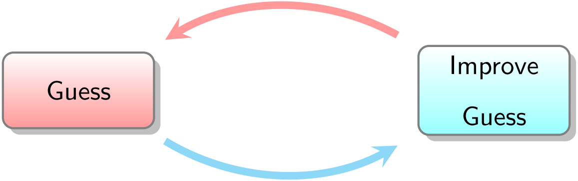
Naive Solution

- Compute value function for all the policies
- Find the best value function
- Find the best policy from the best value function

Classic Solution Approaches



Value Iteration



Simple Puzzle

- Solve for $x^* = g(x^*)$, where

$$g(x) = \frac{1}{2}\left(x + \frac{\alpha}{x}\right)$$

- x^* is called fixed point

Make Guess and Improve the Guess

- Start with initial guess x_0

Make Guess and Improve the Guess

- Start with initial guess x_0
- See if x_0 satisfies the equation

Make Guess and Improve the Guess

- Start with initial guess x_0
- See if x_0 satisfies the equation
- If not let $x_1 = g(x_0)$

Make Guess and Improve the Guess

- Start with initial guess x_0
- See if x_0 satisfies the equation
- If not let $x_1 = g(x_0)$
- Now check if x_1 satisfies the answer otherwise repeat

Make Guess and Improve the Guess

- Start with initial guess x_0
- See if x_0 satisfies the equation
- If not let $x_1 = g(x_0)$
- Now check if x_1 satisfies the answer otherwise repeat
- Stop when iterates do not change much

Example: Make Guess and Improve

- Let $\alpha = 16$ in our problem

Example: Make Guess and Improve

- Let $\alpha = 16$ in our problem
- Let us start with $x_0 = 20$

Example: Make Guess and Improve

- Let $\alpha = 16$ in our problem
- Let us start with $x_0 = 20$
- $x_1 = 10.4$, $x_2 = 5.96$, $x_3 = 4.32$, $x_4 = 4.01$, $x_5 = 4.00$, $x_6 = 4.00$

Example: Make Guess and Improve

- Let $\alpha = 16$ in our problem
- Let us start with $x_0 = 20$
- $x_1 = 10.4$, $x_2 = 5.96$, $x_3 = 4.32$, $x_4 = 4.01$, $x_5 = 4.00$, $x_6 = 4.00$
- Final answer $x^* = x_6 = 4.00$

What fixed point equation do we have?

- Bellman Equation starting at state i

$$V^*(i) = \max_{a \in A} \sum_{j \in S} P_{ij}(a) \ \left[R(i,a,j) + \gamma V^*(j)\right]$$

Cost from i $=$ Immediate Cost $+$ Future Cost from other states

What fixed point equation do we have?

- Bellman Equation starting at state i

$$V^*(i) = \max_{a \in A} \sum_{j \in S} P_{ij}(a) \left[R(i, a, j) + \gamma V^*(j) \right]$$

$$\text{Cost from } i \;=\; \text{Immediate Cost} + \; \text{Future Cost from other states}$$

- Optimal Value function V^* is fixed point

Value Iteration

- Start with initial guess for value function $V_0 = 0$

Value Iteration

- Start with initial guess for value function $V_0 = 0$
- Evaluate L.H.S of Bellman equation and get our new estimate V_1

Value Iteration

- Start with initial guess for value function $V_0 = 0$
- Evaluate L.H.S of Bellman equation and get our new estimate V_1
- Check if the iterates V_0 and V_1 do not change much

Value Iteration

- Start with initial guess for value function $V_0 = 0$
- Evaluate L.H.S of Bellman equation and get our new estimate V_1
- Check if the iterates V_0 and V_1 do not change much
- Repeat until the iterates V_n and V_{n+1} do not change much

Evaluating L.H.S. of Bellman Equation

- For each action a compute the expected long-run reward $TC(i,a)$ starting from i

Evaluating L.H.S. of Bellman Equation

- For each action a compute the expected long-run reward $TC(i, a)$ starting from i
- $TC(i, a)$ is the sum of expected immediate reward for action a and discounted expected future reward

Evaluating L.H.S. of Bellman Equation

- For each action a compute the expected long-run reward $TC(i, a)$ starting from i
- $TC(i, a)$ is the sum of expected immediate reward for action a and discounted expected future reward
- Expected immediate reward $\sum_j P_{ij}(a)\ R(i, a, j)$

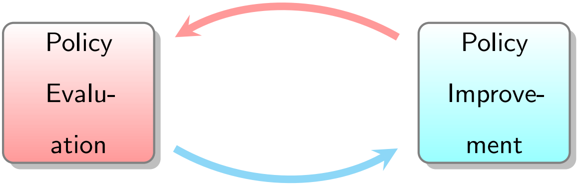
Evaluating L.H.S. of Bellman Equation

- For each action a compute the expected long-run reward $TC(i, a)$ starting from i
- $TC(i, a)$ is the sum of expected immediate reward for action a and discounted expected future reward
- Expected immediate reward $\sum_j P_{ij}(a)\ R(i, a, j)$
- Exected future reward comes our guess, here V_0. i.e., $\sum_j P_{ij}(a)\ V_0(j)$

Evaluating L.H.S. of Bellman Equation

- For each action a compute the expected long-run reward $TC(i, a)$ starting from i
- $TC(i, a)$ is the sum of expected immediate reward for action a and discounted expected future reward
- Expected immediate reward $\sum_j P_{ij}(a)\ R(i, a, j)$
- Exected future reward comes our guess, here V_0. i.e., $\sum_j P_{ij}(a)\ V_0(j)$
- Now $V_1(i) = \max_a\ TC(i, a)$

Policy Iteration



Steps in Policy Iteration

1. Start with a policy say μ

Steps in Policy Iteration

1. Start with a policy say μ
2. Policy Evaluation

Steps in Policy Iteration

1. Start with a policy say μ
2. Policy Evaluation
   - Determine the state-action value function for the policy μ

Steps in Policy Iteration

1. Start with a policy say μ
2. Policy Evaluation
   - Determine the state-action value function for the policy μ
3. Policy Improvement

Steps in Policy Iteration

1. Start with a policy say μ
2. Policy Evaluation
   - Determine the state-action value function for the policy μ
3. Policy Improvement
   - Find $\bar{\mu}$ better than μ

Steps in Policy Iteration

1. Start with a policy say μ
2. Policy Evaluation
   - Determine the state-action value function for the policy μ
3. Policy Improvement
   - Find $\bar{\mu}$ better than μ
   - $V^{\bar{\mu}} \geq V^{\mu}$

Steps in Policy Iteration

1. Start with a policy say μ
2. Policy Evaluation
   - Determine the state-action value function for the policy μ
3. Policy Improvement
   - Find $\bar{\mu}$ better than μ
   - $V^{\bar{\mu}} \geq V^{\mu}$
4. Iterate with the new policy $\bar{\mu}$

Q-values or state-action value function

- Quality of taking action a in state i and then following policy μ

$$Q^{\mu}(i,a) = \max_{a \in A} \sum_{j \in S} P_{ij}(a) \left[R(i,a,j) + \gamma V^{\mu}(j) \right] \; \forall i \in S. \tag{7}$$

- How to compute $Q^{\mu}(i,a)$?

Policy Evaluation

- Compute value function V^μ
  - Either by solving system of linear equations
  - Using value iteration for a fixed policy
- Compute $Q^\mu(i, a)$ from V^μ and the model information P, R

Policy Evaluation: Solving Linear System of Equations

- Get a closed form expression for V^μ

$$
\begin{aligned}
V^\mu &= R^\mu + \gamma P^\mu V^\mu \\
(I - \gamma P^\mu) V^\mu &= R^\mu \\
V^\mu &= (I - \gamma P^\mu)^{-1} R^\mu
\end{aligned}
$$

Policy Evaluation: Value Iteration for a fixed policy

- Start with initial $V_0 = 0$
- Repeatedly apply the function and get a better estimate

$$V_{n+1} = R^{\mu} + \gamma P^{\mu} V_n, \tag{8}$$

- Stop when there is no significant change between consecutive estimates

Policy Improvement

- Find better policy $\mu^{'}$ than μ
- $\bar{\mu} = \max \; Q^{\mu}(i, a)$
- $V^{\mu} \geq V^{\mu}$

Summary

$\boxed{\text{Markov Decision Process (MDP)}}$

Summary

Markov Decision Process (MDP)

Model based Methods

Dynamic Programming (DP)

Value Iteration

Policy Iteration

Summary

Markov Decision Process (MDP)

Model based Methods

Dynamic Programming (DP)

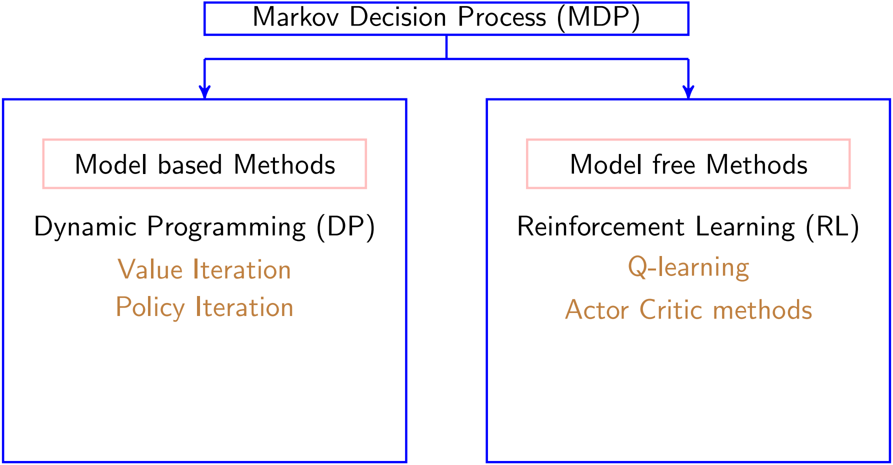
Value Iteration

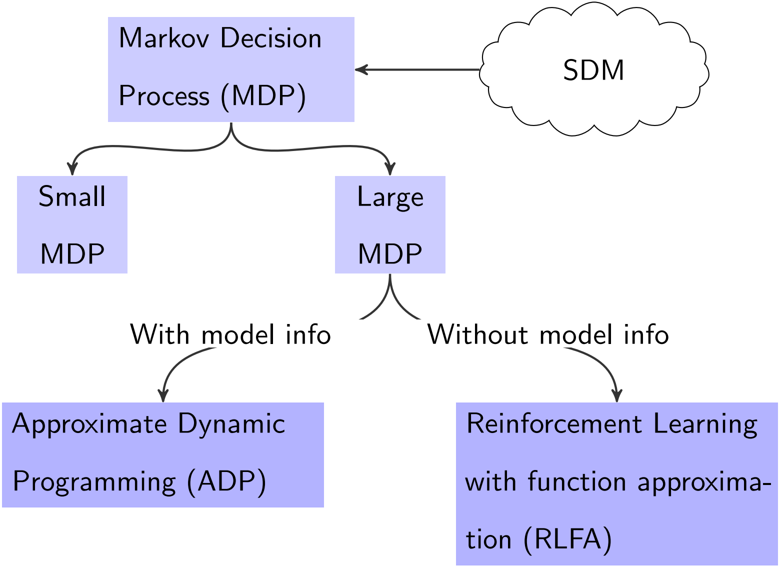
Policy Iteration

Model free Methods

Reinforcement Learning (RL)

Summary


Markov Decision Process (MDP)
Model based Methods
Dynamic Programming (DP)
Value Iteration
Policy Iteration
Model free Methods
Reinforcement Learning (RL)
Q-learning
Actor Critic methods


Markov Decision Process (MDP)
SDM
Small MDP
Large MDP
With model info
Without model info
Approximate Dynamic Programming (ADP)
Reinforcement Learning with function approximation (RLFA)

Questions

Thank you !