Markov Decision Processes and Policy Iteration

Last Time

- What does "**Markov**" mean in "Markov Process"?
- What is a **Markov decision process**?

Guiding Questions

Guiding Questions

- What is a **Markov decision process**?

Guiding Questions

- What is a **Markov decision process**?
- What is a **policy**?

Guiding Questions

- What is a **Markov decision process**?
- What is a **policy**?
- How do we **evaluate** policies?

MDP "Tuple Definition"

MDP "Tuple Definition"

(S, A, T, R, γ)

MDP "Tuple Definition"

(S, A, T, R, γ) (and b and/or S_T in some contexts)

MDP "Tuple Definition"

(S, A, T, R, γ) (and b and/or S_T in some contexts)

- S (state space) - set of all possible states

MDP "Tuple Definition"

(S, A, T, R, γ) (and b and/or S_T in some contexts)

$\{1, 2, 3\}$

- S (state space) - set of all possible states

MDP "Tuple Definition"

(S, A, T, R, γ) (and b and/or S_T in some contexts)

- S (state space) - set of all possible states

$\{1, 2, 3\}$

$\{\text{healthy}, \text{pre-cancer}, \text{cancer}\}$

MDP "Tuple Definition"

(S, A, T, R, γ) (and b and/or S_T in some contexts)

$\{1, 2, 3\}$ $\mathbb{R}^2$

- S (state space) - set of all possible states

$\{\text{healthy}, \text{pre-cancer}, \text{cancer}\}$

MDP "Tuple Definition"

(S, A, T, R, γ) (and b and/or S_T in some contexts)

- S (state space) - set of all possible states

$\{1, 2, 3\}$ $\mathbb{R}^2$ $\{0, 1\} \times \mathbb{R}^4$

$\{\text{healthy}, \text{pre-cancer}, \text{cancer}\}$

MDP "Tuple Definition"

(S, A, T, R, γ) (and b and/or S_T in some contexts)

- S (state space) - set of all possible states

$\{1, 2, 3\}$ $\quad (x, y) \in \mathbb{R}^2$ $\quad \{0, 1\} \times \mathbb{R}^4$

$\{\text{healthy}, \text{pre-cancer}, \text{cancer}\}$

MDP "Tuple Definition"

(S, A, T, R, γ) (and b and/or S_T in some contexts)

- S (state space) - set of all possible states

$\{1, 2, 3\}$ $(x, y) \in \mathbb{R}^2$ $\{0, 1\} \times \mathbb{R}^4$

$\{\text{healthy}, \text{pre-cancer}, \text{cancer}\}$ $(s, i, r) \in \mathbb{N}^3$

MDP "Tuple Definition"

(S, A, T, R, γ) (and b and/or S_T in some contexts)

$\{1, 2, 3\}$ $(x, y) \in \mathbb{R}^2$ $\{0, 1\} \times \mathbb{R}^4$

- S (state space) - set of all possible states

$\{\text{healthy}, \text{pre-cancer}, \text{cancer}\}$ $(s, i, r) \in \mathbb{N}^3$

- A (action space) - set of all possible actions

MDP "Tuple Definition"

(S, A, T, R, γ) (and b and/or S_T in some contexts)

- S (state space) - set of all possible states

$\{1, 2, 3\}$ $(x, y) \in \mathbb{R}^2$ $\{0, 1\} \times \mathbb{R}^4$

$\{\text{healthy}, \text{pre-cancer}, \text{cancer}\}$ $(s, i, r) \in \mathbb{N}^3$

- A (action space) - set of all possible actions

$\{1, 2, 3\}$

MDP "Tuple Definition"

(S, A, T, R, γ) (and b and/or S_T in some contexts)

- S (state space) - set of all possible states

$\{1, 2, 3\}$ $(x, y) \in \mathbb{R}^2$ $\{0, 1\} \times \mathbb{R}^4$

$\{\text{healthy}, \text{pre-cancer}, \text{cancer}\}$ $(s, i, r) \in \mathbb{N}^3$

- A (action space) - set of all possible actions

$\{1, 2, 3\}$

$\{\text{test}, \text{wait}, \text{treat}\}$

MDP "Tuple Definition"

(S, A, T, R, γ) (and b and/or S_T in some contexts)

- S (state space) - set of all possible states

$\{1, 2, 3\}$ $(x, y) \in \mathbb{R}^2$ $\{0, 1\} \times \mathbb{R}^4$

$\{\text{healthy}, \text{pre-cancer}, \text{cancer}\}$ $(s, i, r) \in \mathbb{N}^3$

- A (action space) - set of all possible actions

$\{1, 2, 3\}$ $\mathbb{R}^2$

$\{\text{test}, \text{wait}, \text{treat}\}$

MDP "Tuple Definition"

(S, A, T, R, γ) (and b and/or S_T in some contexts)

$\{1, 2, 3\}$ $(x, y) \in \mathbb{R}^2$ $\{0, 1\} \times \mathbb{R}^4$

- S (state space) - set of all possible states

$\{\text{healthy}, \text{pre-cancer}, \text{cancer}\}$ $(s, i, r) \in \mathbb{N}^3$

- A (action space) - set of all possible actions

$\{1, 2, 3\}$ $\mathbb{R}^2$ $\{0, 1\} \times \mathbb{R}^2$

$\{\text{test}, \text{wait}, \text{treat}\}$

MDP "Tuple Definition"

(S, A, T, R, γ) (and b and/or S_T in some contexts)

- S (state space) - set of all possible states

  $\{1, 2, 3\}$ $(x, y) \in \mathbb{R}^2$ $\{0, 1\} \times \mathbb{R}^4$

  $\{\text{healthy}, \text{pre-cancer}, \text{cancer}\}$ $(s, i, r) \in \mathbb{N}^3$

- A (action space) - set of all possible actions

  $\{1, 2, 3\}$ $\mathbb{R}^2$ $\{0, 1\} \times \mathbb{R}^2$

  $\{\text{test}, \text{wait}, \text{treat}\}$

- T (transition distribution) - explicit or implicit ("generative") model of how the state changes

MDP "Tuple Definition"

(S, A, T, R, γ) (and b and/or S_T in some contexts)

- S (state space) - set of all possible states

$\{1, 2, 3\}$ $(x, y) \in \mathbb{R}^2$ $\{0, 1\} \times \mathbb{R}^4$

$\{\text{healthy}, \text{pre-cancer}, \text{cancer}\}$ $(s, i, r) \in \mathbb{N}^3$

- A (action space) - set of all possible actions

$\{1, 2, 3\}$ $\mathbb{R}^2$ $\{0, 1\} \times \mathbb{R}^2$

$\{\text{test}, \text{wait}, \text{treat}\}$

- T (transition distribution) - explicit or implicit ("generative") model of how the state changes

$T(s' \mid s, a)$

MDP "Tuple Definition"

(S, A, T, R, γ) (and b and/or S_T in some contexts)

- S (state space) - set of all possible states

$\{1, 2, 3\}$ $(x, y) \in \mathbb{R}^2$ $\{0, 1\} \times \mathbb{R}^4$

$\{\text{healthy}, \text{pre-cancer}, \text{cancer}\}$ $(s, i, r) \in \mathbb{N}^3$

- A (action space) - set of all possible actions

$\{1, 2, 3\}$ $\mathbb{R}^2$ $\{0, 1\} \times \mathbb{R}^2$

$\{\text{test}, \text{wait}, \text{treat}\}$

- T (transition distribution) - explicit or implicit ("generative") model of how the state changes

$T(s' \mid s, a)$

- R (reward function) - maps each state and action to a reward

MDP "Tuple Definition"

(S, A, T, R, γ) (and b and/or S_T in some contexts)

- S (state space) - set of all possible states

$\{1, 2, 3\}$ $(x, y) \in \mathbb{R}^2$ $\{0, 1\} \times \mathbb{R}^4$

$\{\text{healthy}, \text{pre-cancer}, \text{cancer}\}$ $(s, i, r) \in \mathbb{N}^3$

- A (action space) - set of all possible actions

$\{1, 2, 3\}$ $\mathbb{R}^2$ $\{0, 1\} \times \mathbb{R}^2$

$\{\text{test}, \text{wait}, \text{treat}\}$

- T (transition distribution) - explicit or implicit ("generative") model of how the state changes

$T(s' \mid s, a)$

- R (reward function) - maps each state and action to a reward

$R(s, a)$ or
$R(s, a, s')$

MDP "Tuple Definition"

(S, A, T, R, γ) (and b and/or S_T in some contexts)

- S (state space) - set of all possible states

$\{1, 2, 3\}$ $(x, y) \in \mathbb{R}^2$ $\{0, 1\} \times \mathbb{R}^4$

$\{\text{healthy}, \text{pre-cancer}, \text{cancer}\}$ $(s, i, r) \in \mathbb{N}^3$

- A (action space) - set of all possible actions

$\{1, 2, 3\}$ $\mathbb{R}^2$ $\{0, 1\} \times \mathbb{R}^2$

$\{\text{test}, \text{wait}, \text{treat}\}$

- T (transition distribution) - explicit or implicit ("generative") model of how the state changes

$T(s' \mid s, a)$

- R (reward function) - maps each state and action to a reward

$R(s, a)$ or

$R(s, a, s')$

$s', r = G(s, a)$

MDP "Tuple Definition"

(S, A, T, R, γ) (and b and/or S_T in some contexts)

- S (state space) - set of all possible states

$\{1, 2, 3\}$ $(x, y) \in \mathbb{R}^2$ $\{0, 1\} \times \mathbb{R}^4$

$\{\text{healthy}, \text{pre-cancer}, \text{cancer}\}$ $(s, i, r) \in \mathbb{N}^3$

- A (action space) - set of all possible actions

$\{1, 2, 3\}$ $\mathbb{R}^2$ $\{0, 1\} \times \mathbb{R}^2$

$\{\text{test}, \text{wait}, \text{treat}\}$

- T (transition distribution) - explicit or implicit ("generative") model of how the state changes

$T(s' \mid s, a)$

- R (reward function) - maps each state and action to a reward

$R(s, a)$ or
$R(s, a, s')$

- γ: discount factor

$s', r = G(s, a)$

MDP "Tuple Definition"

(S, A, T, R, γ) (and b and/or S_T in some contexts)

- S (state space) - set of all possible states

$\{1, 2, 3\}$ $(x, y) \in \mathbb{R}^2$ $\{0, 1\} \times \mathbb{R}^4$

$\{\text{healthy}, \text{pre-cancer}, \text{cancer}\}$ $(s, i, r) \in \mathbb{N}^3$

- A (action space) - set of all possible actions

$\{1, 2, 3\}$ $\mathbb{R}^2$ $\{0, 1\} \times \mathbb{R}^2$

$\{\text{test}, \text{wait}, \text{treat}\}$

- T (transition distribution) - explicit or implicit ("generative") model of how the state changes

$T(s' \mid s, a)$

- R (reward function) - maps each state and action to a reward

$R(s, a)$ or $R(s, a, s')$

- γ: discount factor

$s', r = G(s, a)$

- b: initial state distribution
- S_t: set of terminal states

Decision Networks and MDPs

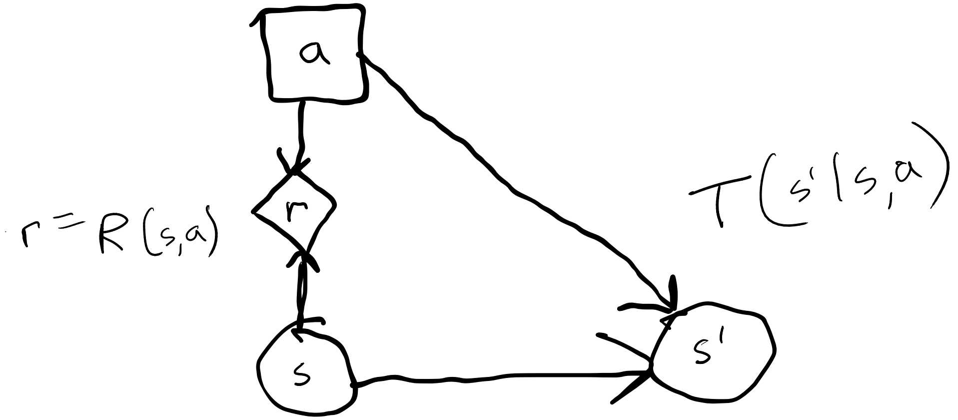
Decision Network

○ Chance node

□ Decision node

◇ Utility node

MDP Dynamic Decision Network



S = set all values that s and s' can take

A = " " " a can take

T

R

γ

MDP Example

Imagine it's a cold day and you're ready to go to work. You have to decide whether to bike or drive.

MDP Example

Imagine it's a cold day and you're ready to go to work. You have to decide whether to bike or drive.

- If you drive, you will have to pay $15 for parking; biking is free.

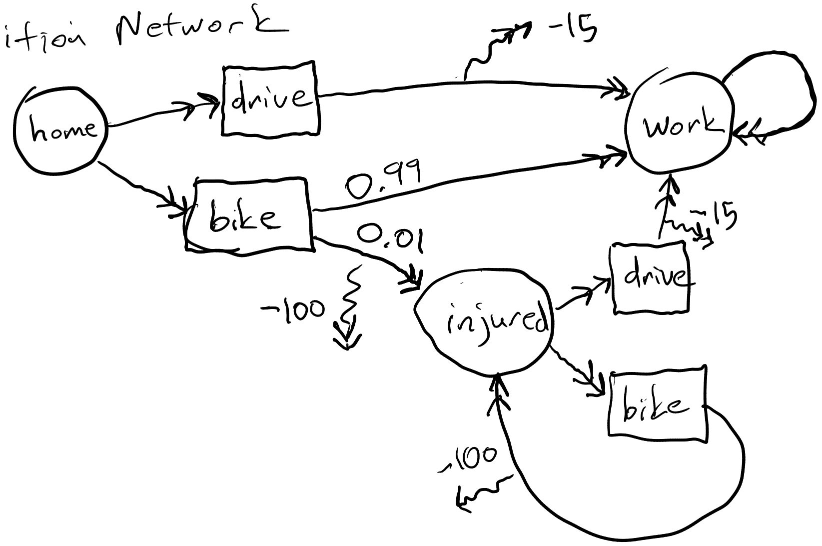
MDP Example

Imagine it's a cold day and you're ready to go to work. You have to decide whether to bike or drive.

- If you drive, you will have to pay \$15 for parking; biking is free.
- On 1% of cold days, the ground is covered in ice and you will crash if you bike, but you can't discover this until you start riding. After your crash, you limp home with pain equivalent to losing \$100.

Not a Decision Network

Transition Network



$S = \{\text{home}, \text{injured}, \text{work}\}$

$A = \{\text{drive}, \text{bike}\}$

$$T^{\text{drive}} = \begin{array}{cc} & \begin{array}{ccc} h & \overset{s'}{i} & w \end{array} \\ \begin{array}{c} h \\ s \;\; i \\ w \end{array} & \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \end{array} \qquad T^{\text{bike}} = \begin{bmatrix} 0 & 0.01 & 0.99 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R(s,a,s') = \begin{cases} -15 & \text{if } a = \text{drive} \\ -100 & \text{if } s' = \text{injured} \\ 0 & \text{o.w.} \end{cases} \qquad \gamma = 0.99$$

$S_T = \{\text{work}\}$ $\quad b(s) \begin{cases} 1 & \text{if } s = \text{home} \\ 0 & \text{o.w.} \end{cases}$

Policies and Simulation

cf. Alg. 9.1, p. 184

Policies and Simulation

- A *policy*, denoted with $\pi(a_t \mid s_t)$, is a conditional distribution of actions given states.

cf. Alg. 9.1, p. 184

Policies and Simulation

- A *policy*, denoted with $\pi(a_t \mid s_t)$, is a conditional distribution of actions given states.
- $a_t = \pi(s_t)$ is used as shorthand when a policy is deterministic.

cf. Alg. 9.1, p. 184

Policies and Simulation

- A *policy*, denoted with $\pi(a_t \mid s_t)$, is a conditional distribution of actions given states.
- $a_t = \pi(s_t)$ is used as shorthand when a policy is deterministic.
- When a policy is combined with an MDP, it becomes a Markov stochastic process with

$$P(s' \mid s) = \sum_a T(s' \mid s, a)\, \pi(a \mid s)$$

cf. Alg. 9.1, p. 184

Policies and Simulation

- A *policy*, denoted with $\pi(a_t \mid s_t)$, is a conditional distribution of actions given states.
- $a_t = \pi(s_t)$ is used as shorthand when a policy is deterministic.
- When a policy is combined with an MDP, it becomes a Markov stochastic process with

$$P(s' \mid s) = \sum_a T(s' \mid s, a)\,\pi(a \mid s)$$

MDP Objective:

$$U(\pi) = \mathrm{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid \pi\right]$$

cf. Alg. 9.1, p. 184

Policies and Simulation

- A *policy*, denoted with $\pi(a_t \mid s_t)$, is a conditional distribution of actions given states.
- $a_t = \pi(s_t)$ is used as shorthand when a policy is deterministic.
- When a policy is combined with an MDP, it becomes a Markov stochastic process with

$$P(s' \mid s) = \sum_a T(s' \mid s, a)\, \pi(a \mid s)$$

MDP Objective:

$$\underline{U(\pi)} = \mathrm{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid \pi\right]$$

Algorithm: Rollout Simulation

Inputs: MDP (S, A, R, T, γ, b) (only need generative model, G), Policy π, horizon H

Outputs: Utility estimate $\hat{u}$

```
$s \leftarrow \mathrm{sample}(b)$
$\hat{u} \leftarrow 0$
for $t$ in $0 \ldots H - 1$
    $a \leftarrow \mathrm{sample}(\pi(a \mid s))$
    $s', r \leftarrow G(s, a)$
    $\hat{u} \leftarrow \hat{u} + \gamma^t r$
    $s \leftarrow s'$
return $\hat{u}$
```

cf. Alg. 9.1, p. 184

Policy Evaluation

Naive Policy Evaluation not on Exam

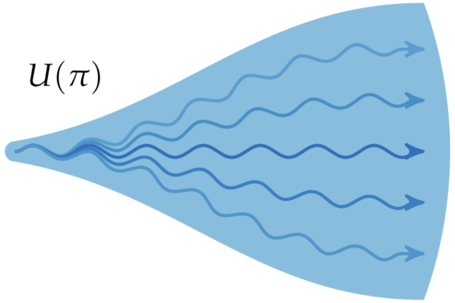
Monte Carlo Policy Evaluation

Monte Carlo Policy Evaluation

- Running a large number of simulations and averaging the accumulated reward is called *Monte Carlo Evaluation*

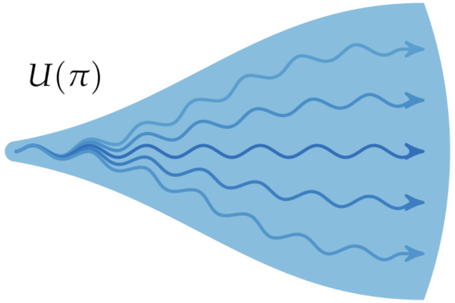
Monte Carlo Policy Evaluation

- Running a large number of simulations and averaging the accumulated reward is called *Monte Carlo Evaluation*



Monte Carlo Policy Evaluation

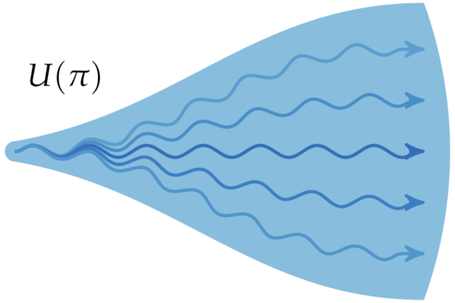
- Running a large number of simulations and averaging the accumulated reward is called *Monte Carlo Evaluation*



Let $\tau = (s_0, a_0, r_0, s_1, \ldots, s_T)$ be a *trajectory* of the MDP

Monte Carlo Policy Evaluation

- Running a large number of simulations and averaging the accumulated reward is called *Monte Carlo Evaluation*

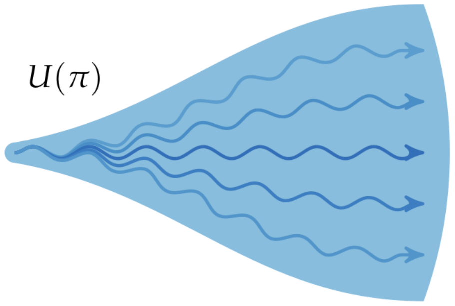


Let $\tau = (s_0, a_0, r_0, s_1, \ldots, s_T)$ be a *trajectory* of the MDP

$$U(\pi) \approx \frac{1}{m} \sum_{i=1}^{m} R(\tau^{(i)})$$

Monte Carlo Policy Evaluation

- Running a large number of simulations and averaging the accumulated reward is called *Monte Carlo Evaluation*



Let $\tau = (s_0, a_0, r_0, s_1, \ldots, s_T)$ be a *trajectory* of the MDP

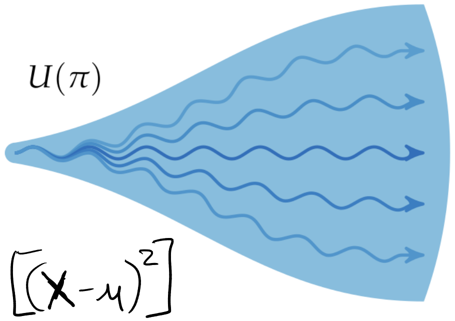
$$U(\pi) \approx \frac{1}{m}\sum_{i=1}^{m} R(\tau^{(i)})$$

$$U(\pi) \approx \overset{\text{R.V. itself}}{\bar{u}_m} = \frac{1}{m}\sum_{i=1}^{m} \hat{u}^{(i)}$$

where $\hat{u}^{(i)}$ is generated by a rollout simulation

Monte Carlo Policy Evaluation

- Running a large number of simulations and averaging the accumulated reward is called *Monte Carlo Evaluation*



$\mu = E[X]$

σ_X = standard dev

$\sigma_X^2 = Var(X) = E[(X-\mu)^2]$

Let $\tau = (s_0, a_0, r_0, s_1, \ldots, s_T)$ be a *trajectory* of the MDP

$$U(\pi) \approx \frac{1}{m}\sum_{i=1}^{m} R(\tau^{(i)})$$

$Var(\bar{u}_m) = \bar{\sigma}_m^2$

$Var(\hat{u}) = \hat{\sigma}^2$

$$U(\pi) \approx \bar{u}_m = \frac{1}{m}\sum_{i=1}^{m} \hat{u}^{(i)}$$

where $\hat{u}^{(i)}$ is generated by a rollout simulation

How can we quantify the accuracy of $\bar{u}_m$?

$$\begin{aligned} Var(\bar{u}_m) &= Var\left(\frac{1}{m}\sum_i \hat{u}^{(i)}\right) \\ &= \frac{1}{m^2} Var\left(\sum_i \hat{u}^{(i)}\right) \\ &= \frac{1}{m^2}\sum_{i=1}^{m} Var(\hat{u}^{(i)}) \quad \text{(Bienaymé)} \end{aligned}$$

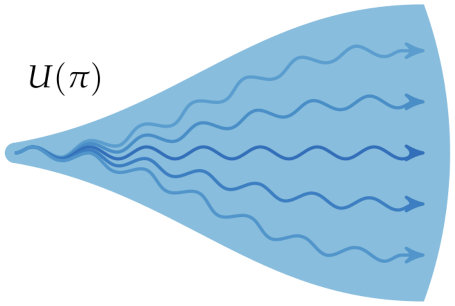
$\bar{\sigma}_m^2 = \frac{1}{m^2} m \hat{\sigma}^2 \Rightarrow \bar{\sigma} = \frac{\hat{\sigma}}{\sqrt{m}}$

Standard Error of Mean

$S.E.M. \equiv \frac{std(\hat{u})}{\sqrt{m}}$

Monte Carlo Policy Evaluation

- Running a large number of simulations and averaging the accumulated reward is called *Monte Carlo Evaluation*



Let $\tau = (s_0, a_0, r_0, s_1, \dots, s_T)$ be a *trajectory* of the MDP

$$U(\pi) \approx \frac{1}{m} \sum_{i=1}^{m} R(\tau^{(i)})$$

How can we quantify the accuracy of $\bar{u}_m$?

also an R.V.

$$U(\pi) \approx \bar{u}_m = \frac{1}{m} \sum_{i=1}^{m} \hat{u}^{(i)}$$

where $\hat{u}^{(i)}$ is generated by a rollout simulation

Value Function-Based Policy Evaluation

Discrete, finite, state and action spaces

$$U^\pi(s) = E\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid \pi, s_0 = s\right]$$

$$U(\pi) = \underset{s\sim b}{E}\left[U^\pi(s)\right]$$

$$= E\left[r_0 \mid s_0 = s, \pi\right] + E\left[\sum_{t=1}^{\infty} \gamma^t r_t \mid \pi, s_0 = s\right]$$

$$= R(s, \pi(s)) + \sum_{s' \in S} T(s' \mid s, a)\, E\left[\sum_{t=1}^{\infty} \gamma^t r_t \mid \pi, \cancel{s_0 = s}, s_1 = s'\right]$$

$$\tau = t - 1$$

$$= R(s, \pi(s)) + \gamma \sum_{s'} T(s' \mid s, a)\, \underbrace{E\left[\sum_{\tau=0}^{\infty} \gamma^\tau r_\tau \mid \pi, s_0 = s'\right]}_{U^\pi(s')}$$

$$\boxed{U^\pi(s) = R(s, \pi(s)) + \gamma \sum_{s'} T(s' \mid s, a)\, U^\pi(s')}$$

Bellman's Expectation Eq.

$$\vec{U}^\pi \qquad \vec{U}^\pi_{ind(s)} = U^\pi(s)$$

$$\vec{R}^\pi_{ind(s)} = R(s, \pi(s))$$

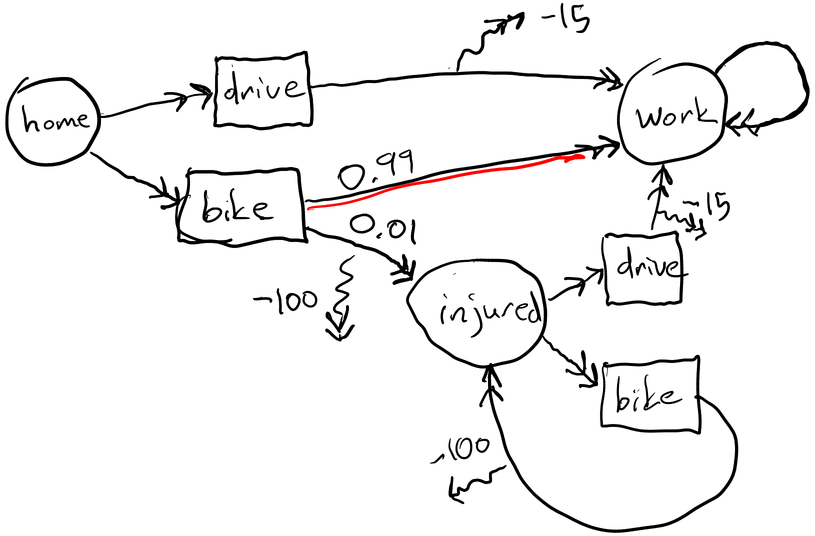
$$T^\pi_{ind(s), ind(s')} = T(s' \mid s, a)$$

$$\vec{U}^\pi = \vec{R}^\pi + \gamma T^\pi \vec{U}^\pi$$

$$\vec{U}^\pi = (I - \gamma T^\pi)^{-1} \vec{R}^\pi$$

Break

- Suggest a policy that you think is optimal for the icy day problem



$$U(\text{drive from home}) = -15$$

$$U(\text{bike from home}) = 0.01 \times (-100 + \overset{\overset{0.9}{\downarrow}}{-15})$$

$$= -1.15$$

$$\pi(s) \begin{cases} \underline{\text{bike}} & \text{if } s = \text{home} \\ \underline{\text{drive}} & \text{if } s = \text{injured} \\ & \end{cases}$$

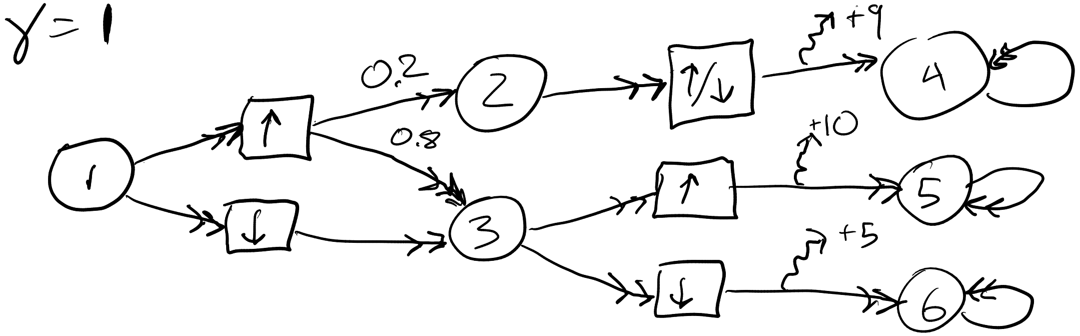
Guiding Questions

Guiding Questions

- How do we reason about the **future consequences** of actions in an MDP?
- What are the basic **algorithms for solving MDPs**?

MDP Example: Up-Down Problem

$\gamma = 1$



$U^*(s) = U^{\pi^*}(s)$

expected sum of future rewards given that we follow the optimal policy

$$U^*(s) = \max_a \left(R(s,a) + \gamma \underset{s' \sim T(s'|s,a)}{E} \left[U^*(s') \right] \right)$$

$$\max_a \left(\underbrace{R(s,a) + \gamma \sum_{s'} T(s'|s,a)\, U^*(s')}_{Q^*(s,a)} \right)$$

$U^*(s) = \max_a Q^*(s,a)$

Bellman Backup Algorithm

$U^*(s) \leftarrow 0$ for all terminal states

Repeat until all $U^*(s)$ are calculated:

find $U^*(s)$ for all states where $U^*(s)$ is known for all children

Extract $\pi^* = \operatorname{argmax} Q^*(s,a)$

$= \operatorname{argmax} \left(R(s,a) + \gamma E[U^*(s)] \right)$

s	a	$Q^*(s,a)$	$U^*(s)$
4	—		0
5	—		0
6	—		0
2	↑/↓	$R(2,\cdot) + (1 \cdot U^*(4))$ $9 + 0 = 9$	9
3	↑ ↓	$R(3,\uparrow) + (1 \cdot U^*(5)) = 10$ $10 + 0$ $R(3,\downarrow) + (1 \cdot U^*(6)) = 5$ $+5 + 0$	10
1	↑ ↓	$R(1,\uparrow) + (0.2 \cdot U^*(2) + 0.8 \cdot U^*(3))$ $0 \quad 0.2 \cdot 9 \quad 0.8 \; 10 = 9.8$ $R(1,\downarrow) + (1 \cdot U^*(3)) = 10$ 0	10

π^*

Dynamic Programming and Value Backup

Dynamic Programming and Value Backup

Bellman's Principle of Optimality: Every sub-policy in an optimal policy is locally optimal

Policy Iteration

Algorithm: Policy Iteration

Given: MDP (S, A, R, T, γ)

Policy Iteration

Algorithm: Policy Iteration

Given: MDP (S, A, R, T, γ)

1. initialize π, π' (differently)

Policy Iteration

```
Algorithm: Policy Iteration
Given: MDP $(S, A, R, T, \gamma)$
    1. initialize $\pi$, $\pi'$ (differently)
    2. while $\pi \neq \pi'$
```

Policy Iteration

Algorithm: Policy Iteration

Given: MDP (S, A, R, T, γ)

```
initialize $\pi$, $\pi'$ (differently)
while $\pi \neq \pi'$
    $\pi \leftarrow \pi'$
```

Policy Iteration

Algorithm: Policy Iteration

Given: MDP (S, A, R, T, γ)

```
1. initialize $\pi$, $\pi'$ (differently)
2. while $\pi \neq \pi'$
3.     $\pi \leftarrow \pi'$
4.     $U^\pi \leftarrow (I - \gamma T^\pi)^{-1} R^\pi$
```

Policy Iteration

Algorithm: Policy Iteration

Given: MDP (S, A, R, T, γ)

```
1. initialize $\pi$, $\pi'$ (differently)
2. while $\pi \neq \pi'$
3.     $\pi \leftarrow \pi'$
4.     $U^\pi \leftarrow (I - \gamma T^\pi)^{-1} R^\pi$
5.     $\pi'(s) \leftarrow \underset{a \in A}{\operatorname{argmax}} \left(R(s,a) + \gamma \sum_{s' \in S} T(s'|s,a) U^\pi(s')\right) \quad \forall s \in S$
```

Policy Iteration

<u>Algorithm: Policy Iteration</u>

Given: MDP (S, A, R, T, γ)

```
initialize $\pi$, $\pi'$ (differently)
while $\pi \neq \pi'$
    $\pi \leftarrow \pi'$
    $U^\pi \leftarrow (I - \gamma T^\pi)^{-1} R^\pi$
    $\pi'(s) \leftarrow \underset{a \in A}{\operatorname{argmax}} \left( R(s,a) + \gamma \sum_{s' \in S} T(s' | s, a) U^\pi(s') \right) \quad \forall s \in S$
return $\pi$
```

Policy Iteration

Algorithm: Policy Iteration

Given: MDP (S, A, R, T, γ)

```
1. initialize $\pi$, $\pi'$ (differently)
2. while $\pi \neq \pi'$
3.     $\pi \leftarrow \pi'$
4.     $U^\pi \leftarrow (I - \gamma T^\pi)^{-1} R^\pi$
5.     $\pi'(s) \leftarrow \underset{a \in A}{\text{argmax}} \left( R(s,a) + \gamma \sum_{s' \in S} T(s'|s,a) U^\pi(s') \right) \quad \forall s \in S$
6. return $\pi$
```

Policy Eval

Policy Improvement Step

(Policy iteration notebook)

Value Iteration

Algorithm: Value Iteration

Given: MDP (S, A, R, T, γ), tolerance ϵ

Value Iteration

Algorithm: Value Iteration

Given: MDP (S, A, R, T, γ), tolerance ϵ

1. initialize U, U' (differently)

Value Iteration

Algorithm: Value Iteration

Given: MDP (S, A, R, T, γ), tolerance ϵ

```
1. initialize $U$, $U'$ (differently)
2. while $\|U - U'\|_\infty > \epsilon$
```

Value Iteration

Algorithm: Value Iteration

Given: MDP (S, A, R, T, γ), tolerance ϵ

```
initialize $U$, $U'$ (differently)
while $\|U - U'\|_\infty > \epsilon$
    $U \leftarrow U'$
```

Value Iteration

Algorithm: Value Iteration

Given: MDP (S, A, R, T, γ), tolerance ϵ

```
initialize $U$, $U'$ (differently)
while $\|U - U'\|_\infty > \epsilon$
    $U \leftarrow U'$
    $U'(s) \leftarrow \max_{a \in A} \left(R(s,a) + \gamma \sum_{s' \in S} T(s'|s,a)U(s')\right) \quad \forall s \in S$
```

Value Iteration

Algorithm: Value Iteration

Given: MDP (S, A, R, T, γ), tolerance ϵ

```
1. initialize $U, U'$ (differently)
2. while $\|U - U'\|_\infty > \epsilon$
3.     $U \leftarrow U'$
4.     $U'(s) \leftarrow \max_{a \in A} \left(R(s,a) + \gamma \sum_{s' \in S} T(s'|s,a) U(s')\right) \quad \forall s \in S$
5. return $U'$
```

Value Iteration

Algorithm: Value Iteration

Given: MDP (S, A, R, T, γ), tolerance ϵ

```
initialize $U, U'$ (differently)
while $\|U - U'\|_\infty > \epsilon$
    $U \leftarrow U'$
    $U'(s) \leftarrow \max_{a \in A} \left(R(s,a) + \gamma \sum_{s' \in S} T(s'|s,a) U(s')\right) \quad \forall s \in S$
return $U'$
```

- Returned U' will be close to U^*!

Value Iteration

Algorithm: Value Iteration

Given: MDP (S, A, R, T, γ), tolerance ϵ

```
1. initialize $U, U'$ (differently)
2. while $\|U - U'\|_\infty > \epsilon$
3.     $\underline{U} \leftarrow U'$
4.     $\underline{U'(s) \leftarrow \max_{a \in A} \left(R(s,a) + \gamma \sum_{s' \in S} T(s'|s,a)\overset{\downarrow}{U}(s')\right)}$    $\forall s \in S$
5. return $U'$
```

- Returned U' will be close to U^*!
- π^* is easy to extract: $\pi^*(s) = \arg\max(R(s,a) + \gamma E[U^*(s')])$

Bellman's Equations

Policy Evaluation

$$U^{\pi}(s) = R(s, \pi(s)) + \gamma \mathop{E}_{s'\sim T(s'|s,\pi(s))}\left[U^{\pi}(s')\right]$$

Bellman's Expectation Equation

Bellman Backup

Certificate of Optimality

$$U^{*}(s) = \max_{a}\left(R(s,a) + \gamma \mathop{E}_{s'\sim T(s'|s,a)}\left[U^{*}(s')\right]\right)$$

Bellman's Optimality Equation

Value Iteration

$$\left\{\begin{aligned} U'(s) &= \max_{a} R(s,a) + \gamma \mathop{E}_{s'\sim T(s'|s,a)}\left[U(s')\right] \\ U'(s) &= B[U](s) \end{aligned}\right.$$

Bellman Operator

VI:

```
initialize $U, U'$
while $||U - U'||_\infty > \varepsilon$
    $U \leftarrow U'$
    $U' \leftarrow B[U]$
```

Guiding Questions

Guiding Questions

- How do we reason about the **future consequences** of actions in an MDP?
- What are the basic **algorithms for solving MDPs**?

Guiding Questions

- How do we reason about the **future consequences** of actions in an MDP?
- What are the basic **algorithms for solving MDPs**?

"In any small change he will have to consider only these quantitative indices (or "values") in which all the relevant information is concentrated; and by adjusting the quantities one by one, he can appropriately rearrange his dispositions without having to solve the whole puzzle ab initio, or without needing at any stage to survey it at once in all its ramifications."

-- F. A. Hayek, "The use of knowledge in society", 1945