P9120 - Statistical Learning and Data Mining

Lecture 10 - Reinforcement Learning II

Min Qian

Department of Biostatistics, Columbia University

November 7th, 2024

Outline

Multi-Armed Bandit Recap

- At each time point $t = 1, 2, \ldots,$
  - You (i.e., agent) face K choices of actions $a \in \{1, 2, \ldots, K\}$.
  - You choose action a_t and observe reward Y_t.

- The Reward of an action a, denoted by $Y_t(a)$, follows a fixed by unknown distribution $D(a)$ with expectation $Q^*(a)$, which only depends on a.

- Goal: Sequentially choose actions a_t to maximize the expected total reward, $\sum_{t=1}^{T} E[R_t | A_t = a_t]$, over some time period (say, $T = 1000$ steps).

From Multi-Armed Bandit to Contextual Bandit

Bandit with side info. (i.e., context) that can help make decisions.

Contextual Bandits: some context

Consider the following example:

- We are running a sports news website. Today, there are K big sports related news stories.
- Every time a user visits our set, we must decide then and there which headlines to display to her on the front page.
- The goal is to maximize the number of clicks.

Bob just visited our website!

1. Bob loves the Toronto Raptors.
2. Hates the Los Angeles Clippers.

Which headline should we show to Bob?

Context

Arms

1.

2.

– from S. Randhawa (July 2019)

Contextual Bandit

- At each time point $t = 1, 2, \ldots,$
  - A user (i.e., environment) presents information (i.e., context) $x_t \in \mathcal{X}$
  - You (i.e., agent) face K choices of actions $a \in \{1, 2, \ldots, K\}$.
  - You choose action $A_t \in \{1, 2, \ldots, K\}$ and observe reward R_t.

- For every context-action pair $x \in \mathcal{X}$ and $a \in \{1, \ldots, K\}$, the reward $R_t(x, a)$, follows a fixed by unknown distribution $D(x, a)$ with expectation $Q^*(x, a)$, which depends on both x and a.

- Goal: Sequentially choose actions A_t to maximize the expected total reward, $\sum_{t=1}^{T} E[R_t(A_t)]$, over some time period (say, $T = 1000$ steps).

Formal Contextual Bandit Set-Up

- At each time point $t = 1, 2, \ldots,$
  - $x_t \in \mathcal{X}$: context (e.g., patient information, consumer information)
  - $A_t \in \{1, \ldots, K\}$: action
  - $R_t \in \mathbb{R}$: reward, distribution depends on (x_t, a_t).

- Usually assume $x_t, t = 1, 2, \ldots,$ to be independent (of each other and previous actions).

- Goal: sequentially select actions $(A_1, \ldots, A_T)$ so that after T time steps, we've incurred almost as much reward as the optimal policy π^*, i.e., minimize

$$Regret(T) = \sum_{t=1}^{T} \Big[E(R_t(\pi^*(x_t))) - E(R_t(A_t)) \Big],$$

where $\pi^*(x) = \arg\max_\pi \sum_{t=1}^{T} E[R_t(\pi(x))]$.

Regression based Contextual Bandit Methods

Assume $Y_t|X_t = x, A_t = a$ are i.i.d. for $t = 1, 2, \ldots$

- $Q^*(x, a) = E(Y_t|A_t = a, X_t = x)$, the expected rewards for action $A_t = a$ with context $X_t = x$.

- Warm-up to obtain initial data. At each time point t:
  - Estimate $Q^*(x, a)$ by $\hat{Q}(x, a) = \Phi(x, a)^T\hat{\theta}$ using linear regression (or other nonparametric estimates) based on accrued data.
  - Choose actions using exploitation-exploration methods in bandits based on $\widehat{Q}(x, a)$:
    e.g.: choose $\hat{a} = \hat{\pi}(x) = \arg\max_a \widehat{Q}(x, a)$ w.p. $1 - \epsilon$ and choose a random action w.p. ϵ.

Article Recommendation in Contextual Linear Bandit Setting

Linear Payoff = $x^T \theta$

Users u_1 with age YOUNG and u_2 with age OLD

$$x_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$x_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

[0.1 , 0.6]

Retirement planning wishes vs. reality

[0.5, 0.1]

The Player **Wizarding World of Harry Potter ride may conjure a new path for theme park rides**

[0.6 , 0.1]

Elon Musk: 198,000 Tesla Model 3 Orders Received in 24 Hours

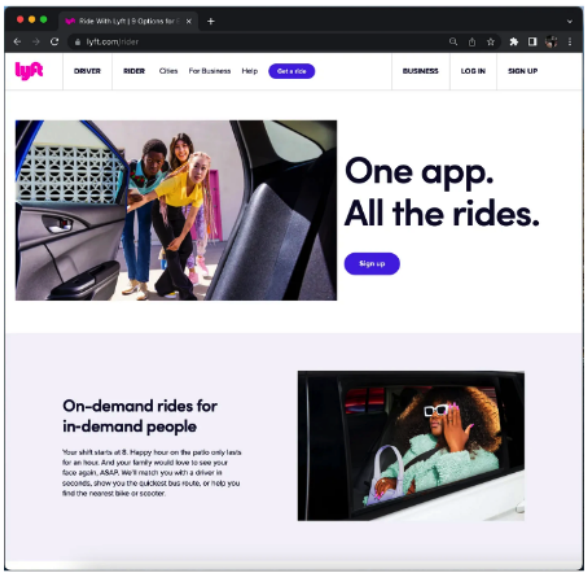
[0.9 , 0.2]

Not tired yet: Warriors top Spurs for 72nd win, set up date with history

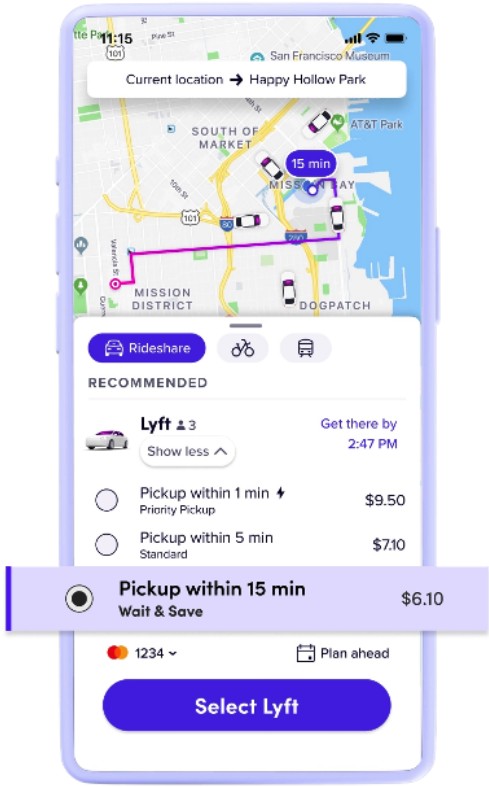
22

MAB to Contextual Bandit: Applications at Lyft

UI optimization: For example, consider landing page for riders on the Lyft website. We can experiment with many different combinations of the value proposition text and main image to find the best combination that inspires riders to take their first ride. To take this a step further, we can use information about the time-of-day or regional information to make the experience adaptive to the context of the visit. Perhaps in the morning, potential riders are more interested in commute options and on the weekends they would prefer to meet up with friends.

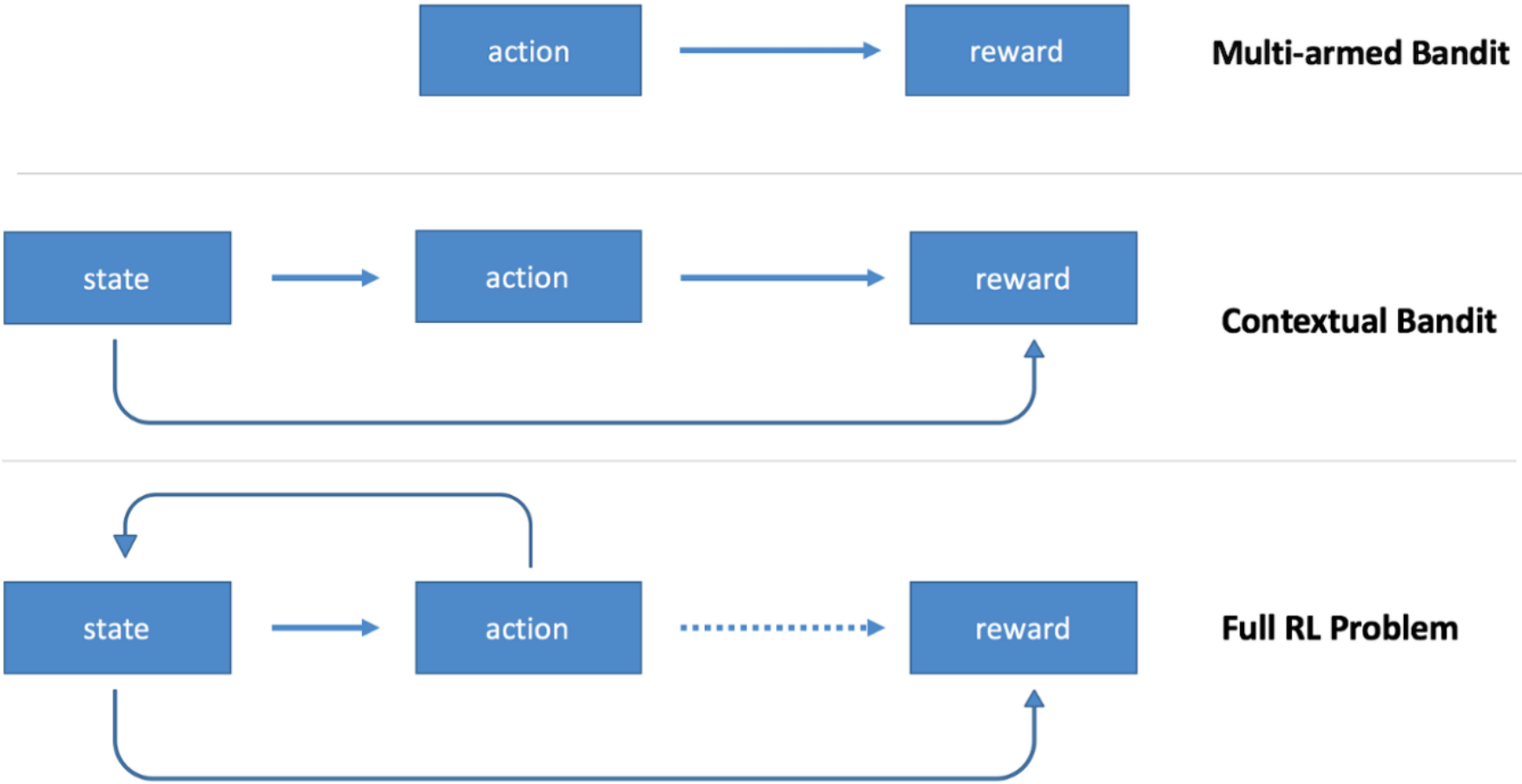


Marketplace optimization: set the pickup window for the Wait and Save offer depending on context such as local ride demand and driver availability. We can choose the window that will provide optimal reliability and cost savings. For this application we employ contextual MAB methods with a reward function that jointly optimizes reliability and ride requests.



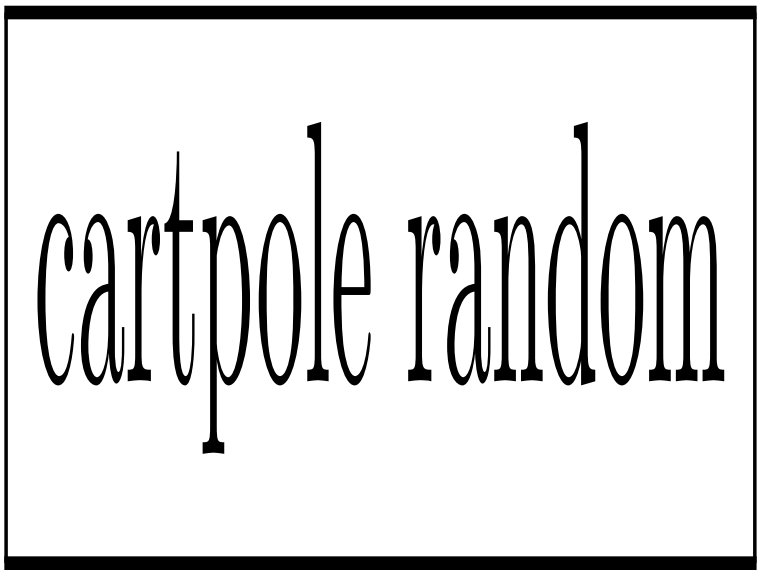
Full Reinforcement Learning

Current context (state, observation) may depend on previous information (state, action, reward in prior time points).



- from "towardsdatascience.com"

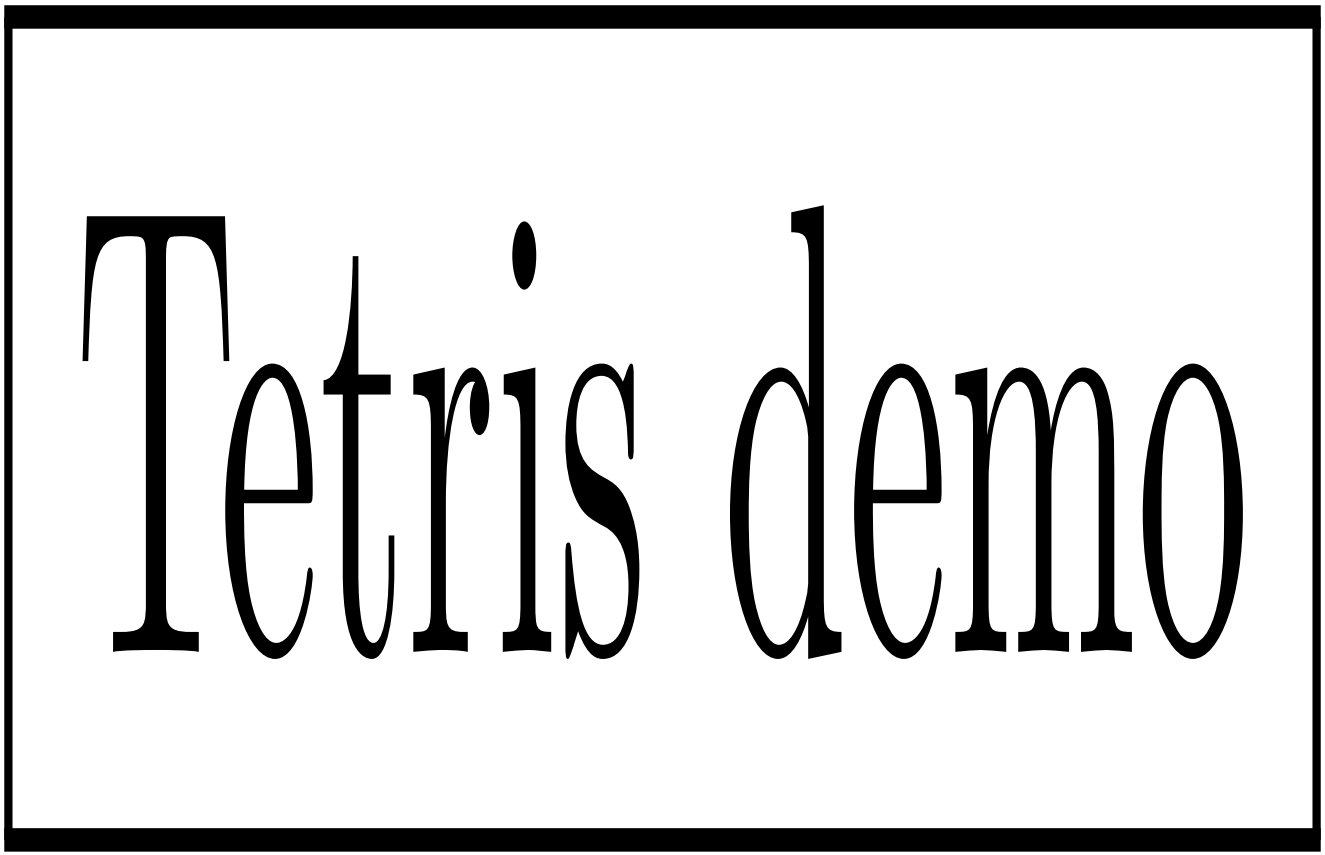
Cart Pole Example



Goal: to keep the pole upright for as long as possible.

- State S: the position/velocity of cart, angle/velocity of the pole.
- Action A: $\{\leftarrow, \rightarrow\}$
- Reward: $R = +1$ for every step the pole has not fallen over, $R = 0$ o.w.

Tetris Example



- State S
- Action A
- Reward R
- Goal

Reinforcement Learning (RL)

- Agent's life

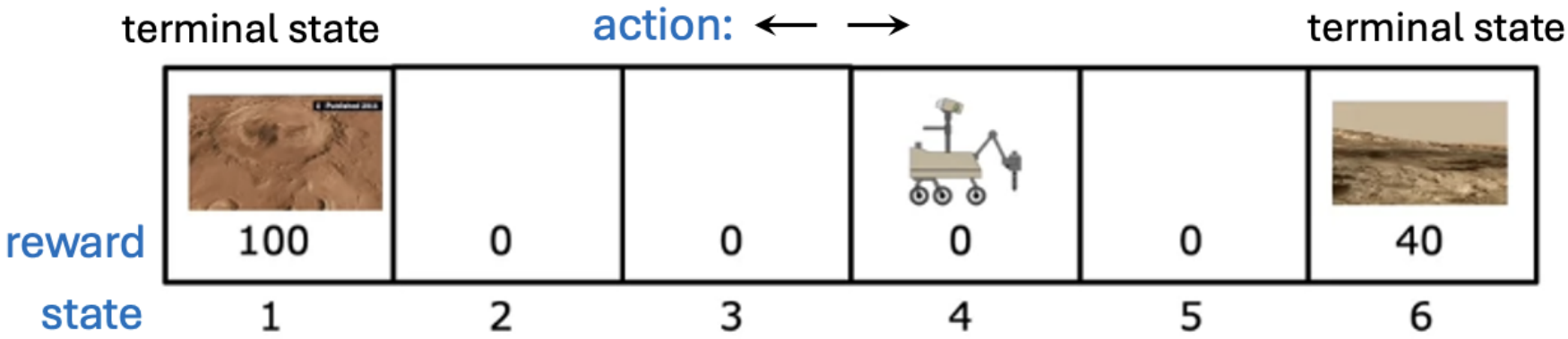
$$X_0, A_0, R_0, X_1, A_1, R_1, \ldots, X_t, A_t, R_t, \ldots$$

- Observations X_t can be vectors or other structures.
- Actions A_t can be multi-dimensional.
- A Reward R_t is a scalar feedback signal

Agent chooses actions so as to maximize the expected cumulative rewards over a time horizon.

Under "Markov" assumption, Observations X_t is replaced by State S_t. In this case, it is called "Markov Decision Process".

Mars Rover Example

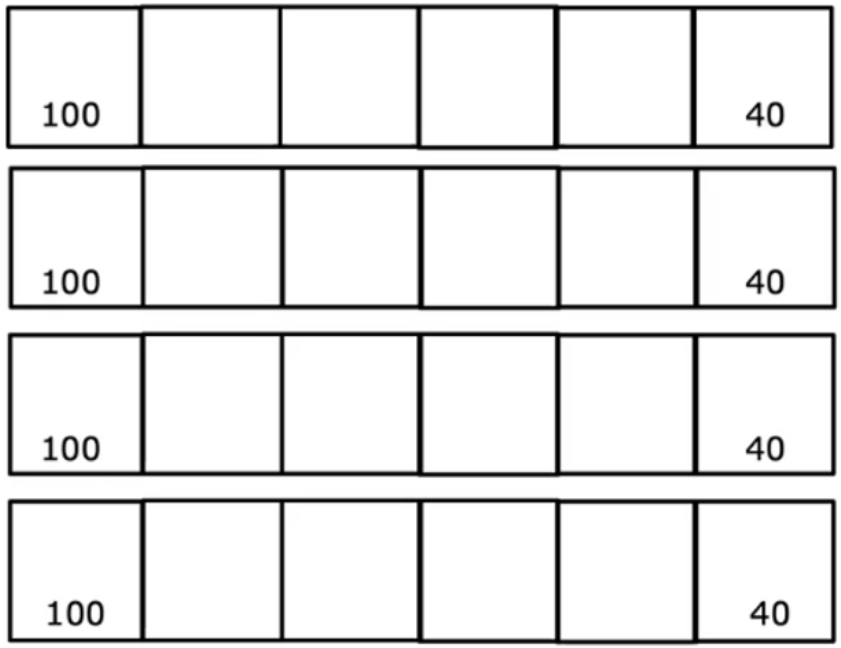


Return at time t: $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k}$ (unified notation)

- $R_t + R_{t+1} + \ldots + R_T$,, where T is a final time.
- $R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \ldots$,, where $\gamma \in [0, 1)$ is a discount factor.

The return depends on the actions you take.

Mars Rover Example: Policy



policy π:

(deterministic) $\pi(s)$:

state $\longrightarrow$ action

(stochastic) $\pi(a|s)$:

probability of taking action a at state s

Goal: Find a policy π that tells you what action to take in every state s, so as to maximize the expected return.

- The expected return following a policy π is *value function of* π.
- The policy that maximizes the expected return is *optimal policy*.
- The value function of the optimal policy is *optimal value function*.

Markov Decision Process (MDP)

$$S_0, A_0, R_0, \ldots, S_t, A_t, R_t, \ldots$$

Markovian (and usually Stationary)

$$\begin{aligned}&\Pr(S_{t+1} = s', R_t = r | S_0, A_0, R_0, \ldots, S_t = s, A_t = a) \\ =&\Pr(S_{t+1} = s', R_t = r | S_t = s, A_t = a) \triangleq p(s', r|s, a)\end{aligned}$$

Finite MDP: The state space contains finite number of states.

- State-transition probabilities
  $p(s'|s,a) \triangleq \Pr(S_{t+1} = s' | S_t = s, A_t = a) = \sum_{r \in \mathcal{R}} p(s', r|s, a)$
- Expected rewards for state-action-next state triplets:
  $r(s,a,s') \triangleq E(R_t | S_t = s, A_t = a, S_{t+1} = s') = \sum_{r \in \mathcal{R}} r \frac{p(s', r|s,a)}{p(s'|s,a)}$
- Expected rewards for state-action pairs:
  $r(s,a) \triangleq E(R_t | S_t = s, A_t = a) = \sum_{s' \in \mathcal{S}} p(s'|s,a) r(s,a,s').$

MDP: Return, Policy, Value

$$S_0, A_0, R_0, \ldots, S_t, A_t, R_t, \ldots$$

- Return (cumulative rewards) $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k}$.
- Policy Deterministic policy $\pi(s) : \mathcal{S} \to \mathcal{A}$;
  Stochastic policy: $\pi(a|s) : \mathcal{S} \times \mathcal{A} \to [0, 1]$;
- (state-)value function: Specifies what is good in a long run.

$$V_\pi(s) = E_\pi(G_0|S_0 = s) = E_\pi(G_t|S_t = s),$$

  i.e., expected return if start in state s and then follow policy π.

- Optimal policy $\pi^*(s) = \arg\max_\pi V_\pi(s)$.
- Optimal value: $V^*(s) = V_{\pi^*}(s)$.
- Goal: Estimate π^*.

There always exists a deterministic optimal policy for an MDP.

Recycling Robot Example

Goal: Recycle as many cans as possible.

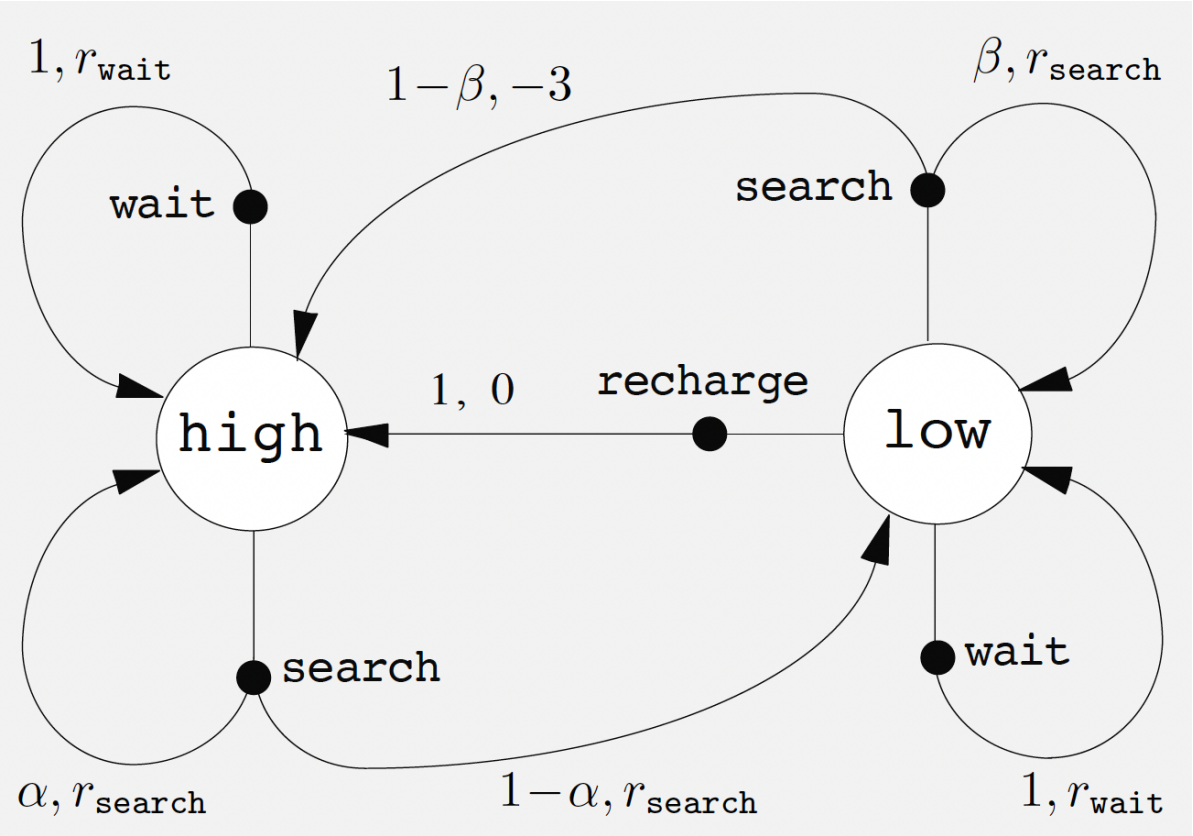
- At each step, robot has to decide whether it should
  - actively search for a can,
  - wait for someone to bring it a can, or
  - go to home base and recharge.
- Searching is better but runs down the battery; if it runs out of power while searching, has to be rescued (which is bad).
- Decisions made on basis of current energy level: high, low
- Reward = number of cans collected, with penalty of running out of power.

System Dynamics of Recycling Robot Example

- $S_t \in \mathcal{S} = \{\text{high, low}\}$
- $A_t \in \mathcal{A}(S_t)$, where
  $\mathcal{A}(\text{high}) = \{\text{search, wait}\}$, $\mathcal{A}(\text{low}) = \{\text{search, wait, recharge}\}$
- R_t : no. of cans collected in period t.

s	a	s'	$p(s' \mid s, a)$	$r(s, a, s')$
high	search	high	α	r_{search}
high	search	low	$1-\alpha$	r_{search}
low	search	high	$1-\beta$	-3
low	search	low	β	r_{search}
high	wait	high	1	r_{wait}
high	wait	low	0	-
low	wait	high	0	-
low	wait	low	1	r_{wait}
low	recharge	high	1	0
low	recharge	low	0	-

Recycling Robot Example



- What's $v_\pi(s)$ for a specific policy π?
- What is the optimal policy π^*?
- What is the optimal value function $V^*(s)$?
- ...

Planning vs Learning

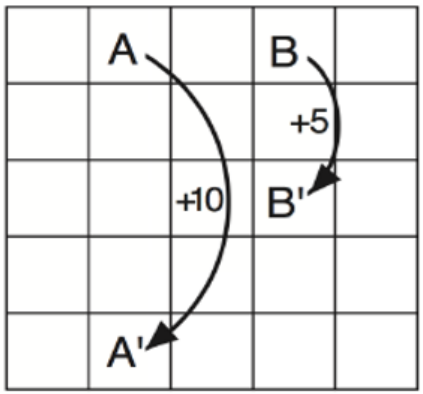
Planning: system dynamics $p(s', r|s, a)$ is known. Develop iterative algorithm to compute quantities of interest, e.g., $V_\pi(s), \pi^*, V_{\pi^*}(s), \ldots$

- Policy evaluation
- Policy improvement
- Policy iteration
- Value iteration
- $\ldots$

Learning: System dynamics $p(s', r|s, a)$ is unknown. Data are available (or there is a blackbox for you to generate data). Develop iterative algorithm to estimate quantities of interest e.g., $V_\pi(s), \pi^*, V_{\pi^*}(s), \ldots$

- TD learning
- Sarsa
- Q-learning
- $\ldots$

Policy Evaluation: Gridworld Example



State space: 25 cells. Action space: $\leftarrow, \rightarrow, \uparrow, \downarrow$

Dynamics of MDP (reward dist. and transition prob.)

$S_{t+1} = S_t, R_t = -1$ if A_t would take the agent off the grid

$S_{t+1} = A', R_t = +10$ if $S_t = \text{A}$ regardless of A_t

$S_{t+1} = B', R_t = +5$ if $S_t = \text{B}$ regardless of A_t

$S_{t+1} = S_t + \textit{one cell in the direction of } A_t, \quad R_t = 0, \text{ o.w.}$

Question: What is the value function of a random policy (i.e., $\pi(a|s) = 1/4$ for all (s, a) pairs) with discount rate $\gamma = 0.9$?

Example: at state B:

$$\begin{aligned} V_\pi(S_0 = B) =& E_\pi(R_0|S_0 = B) \\ &+ \gamma E_\pi(R_1|S_0 = B) \\ &+ \gamma^2 E_\pi(R_2|S_0 = B) + \dots \\ =& 5 + 0 + 0.9(-1/16) + \dots \end{aligned}$$

Bellman Equation for Value Function $V_\pi(s)$

Note that $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k} = R_t + \gamma G_{t+1}$.

$$
\begin{aligned}
V_\pi(s) &\triangleq E_\pi[G_t|S_t = s] = E_\pi\Big[R_t + \gamma G_{t+1}\Big|S_t = s\Big] \\
&= E_\pi\Big[R_t + \gamma E_\pi(G_{t+1}|S_{t+1}, S_t = s)\Big|S_t = s\Big] \\
&= E_\pi\Big[R_t + \gamma V_\pi(S_{t+1})\Big|S_t = s\Big]. \\
&= \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)[r + \gamma V_\pi(s')] \\
&= \sum_a \pi(a|s) \sum_{s'} p(s'|s, a)\Big[r(s, a, s') + \gamma V_\pi(s')\Big] \\
&= \sum_a \pi(a|s)\Big[r(s, a) + \gamma \sum_{s'} p(s'|s, a) V_\pi(s')\Big]
\end{aligned}
$$

$V_\pi(s)$ is the unique solution to a system of linear equations!

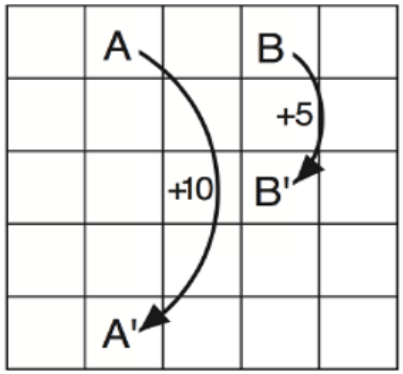
Recycling Robot Example

$$V_\pi(s) = \sum_a \pi(a|s) \sum_{s'} p(s'|s,a) \Big[r(s,a,s') + \gamma V_\pi(s') \Big]$$

s	a	s'	$p(s' \mid s,a)$	$r(s,a,s')$
high	search	high	α	$r_{\texttt{search}}$
high	search	low	$1-\alpha$	$r_{\texttt{search}}$
low	search	high	$1-\beta$	-3
low	search	low	β	$r_{\texttt{search}}$
high	wait	high	1	$r_{\texttt{wait}}$
high	wait	low	0	-
low	wait	high	0	-
low	wait	low	1	$r_{\texttt{wait}}$
low	recharge	high	1	0
low	recharge	low	0	-

Gridworld Example

$$V_\pi(s) = \sum_a \pi(a|s)\left[r(s,a) + \gamma \sum_{s'} p(s'|s,a) V_\pi(s')\right]$$



State space: 25 cells. Action space: $\leftarrow, \rightarrow, \uparrow, \downarrow$

Dynamics of MDP (reward dist. and transition prob.)

$S_{t+1} = S_t, R_t = -1$ if A_t would take the agent off the grid

$S_{t+1} = A', R_t = +10$ if $S_t = \text{A}$ regardless of A_t

$S_{t+1} = B', R_t = +5$ if $S_t = \text{B}$ regardless of A_t

$S_{t+1} = S_t + \textit{one cell in the direction of } A_t, \quad R_t = 0,$ o.w.

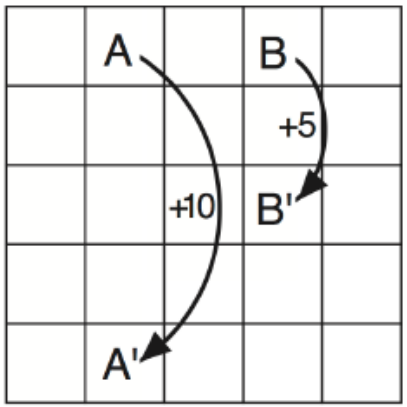
Question: What is the value function of a random policy (i.e., $\pi(a|s) = 1/4$ for all (s,a) pairs) with discount rate $\gamma = 0.9$?

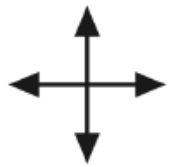
Planning: Policy Evaluation

Iterative Policy Evaluation, for estimating $V \approx v_\pi$

Input π, the policy to be evaluated
Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$ arbitrarily, for $s \in \mathcal{S}$, and $V(terminal)$ to 0

Loop:
  $\Delta \leftarrow 0$
  Loop for each $s \in \mathcal{S}$:
    $v \leftarrow V(s)$
    $V(s) \leftarrow \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)\left[r + \gamma V(s')\right]$
    $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$





3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

Reminders

- Quiz for Lecture 10 is due at 9pm on Monday November 11th.

- In class group paper presentation (next week Nov 14th):
  - Topic: Deep representation learning of electronic health records to unlock patient stratification at scale.
    Team members: Yimeng Cai, Yuxuan Du, Yuki Low, Hyunjee Oh, Haotian Tang, Yang Zhao
  - Each presentation is about 40-45 minutes.
  - Every student in the group is expected to present.
  - Evaluation will be based on both individual performance (80%) and group performance (20%).