MGMT298D
Science and Strategy of AI

Week 3

Reinforcement Learning

Auyon Siddiq
UCLA Anderson School of Management

Poker Chip Experiment

Retailer wants to test three different product pages

20 customers arrive

Each pouch represents one product page

A green poker chip is a successful conversion (e.g., a purchase), and a red is no conversion

How many green can you get?

Today's Class: Reinforcement Learning

How do we make decisions in real-time when faced with uncertainty?

Reinforcement learning is the sub-field of AI focused on this

We will look at two RL frameworks widely used in industry: **multi-armed bandits** and **Q-learning**

Adaptive Experimentation: Alibaba and Amazon

Case Handout

Discussion

The Limits of A/B Testing

Traditional A/B testing:

1) Split units (e.g., users) into fixed *treatment* and *control* groups (e.g., 1% / 10% / 50% treatment depending on context)
2) Compare means of both groups and calculate *p*-value
3) If p-value small, conclude treatment and control are different

Downside is that experimentation is costly! If treatment is terrible, it's a wasted opportunity

Idea: Adapt allocation to "treatment" as we learn more

Multi-Armed Bandits: Dynamic A/B Testing

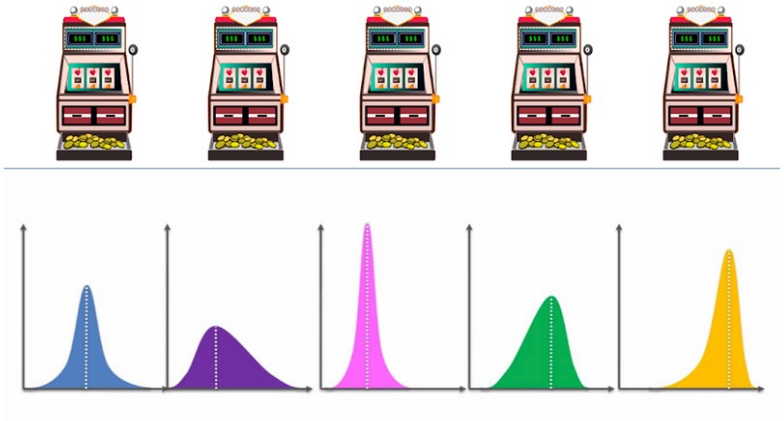
Multi-Armed Bandits: The Framework

The Setup

K arms with unknown reward probabilities

Each step: pull one arm (action), observe reward

Goal: maximize total reward over T rounds



Applications

Web optimization: arms = layouts

Ad selection: arms = ads

Pricing: arms = price levels

Bandits “learn and earn” simultaneously. Each “arm” has an unknown reward distribution. Simplest case is binary rewards with unknown success probability *p*

Key is to manage the exploration-exploitation trade-off

A/B Test vs. Bandits

A/B Test (fixed 25% each)

Bandit (adaptive — after learning)

Bandits shift rounds (e.g., users) toward the best arm gradually

Most useful in settings with hundreds or thousands of rounds (e-commerce, digital platforms, serving ads)

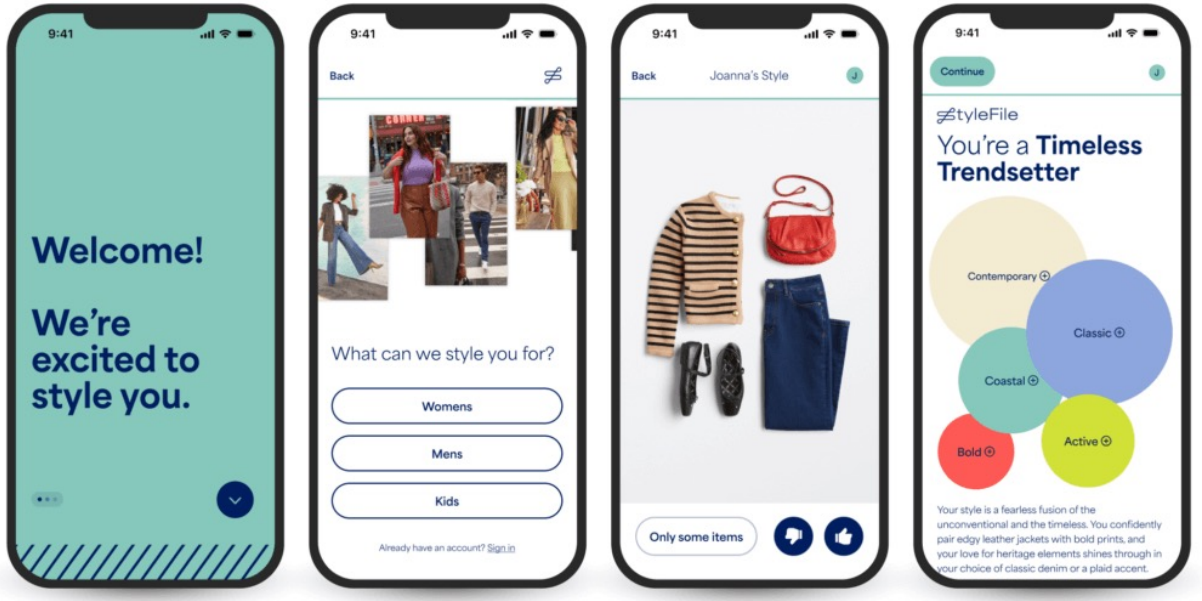
Client Outreach at StitchFix

How to re-engage inactive clients?

A – Do nothing

B – Offer 1-on-1 stylist appointment

C – Offer promotion



https://multithreaded.stitchfix.com/blog/2018/11/08/bandits/

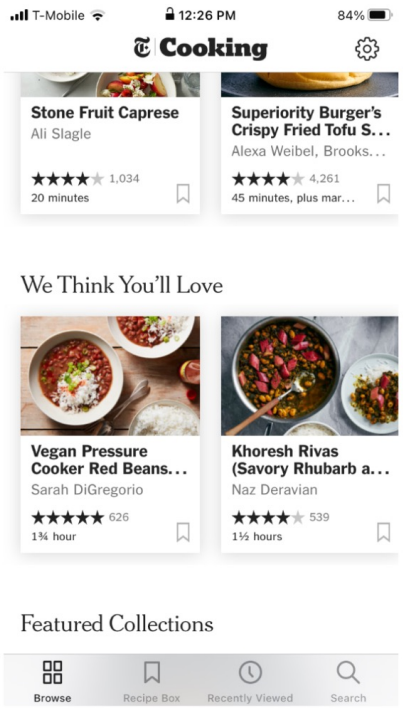
Recipe Recommendations at the New York Times

What recipe to recommend?

A – Stone Fruit Caprese Salad

B – Shaved Fennel and Citrus Salad with Pistachios

C – Charred Corn and Black Bean Salad with Lime Crema, Pickled Red Onion, Cotija, and Whatever Else You Have Lying Around



https://medium.com/@timesopen/how-the-new-york-times-cooking-team-makes-personalized-recipe-recommendations-669c26aa4825

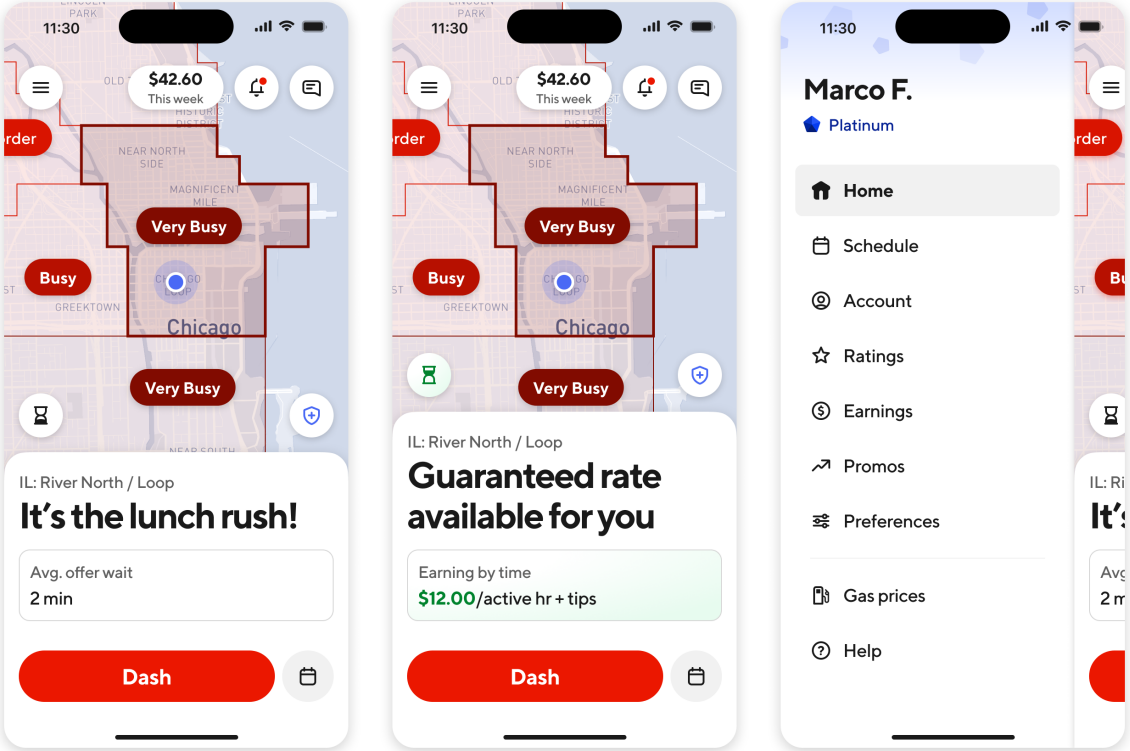
Responsive Drivers at DoorDash

Which offline driver to send notification to?

A – Driver A

B – Driver B

C – Driver C



https://careersatdoordash.com/blog/using-a-multi-armed-bandit-with-thompson-sampling-to-identify-responsive-dashers/

Example: Price Experimentation

Suppose an online retailer is trying to decide between 4 prices for a product:

$29.99 $39.99 $49.99 $59.99

The customer purchase probabilities under each price unknown and must be learned through experimentation.

Which price maximizes revenue?

Strategy 0: Completely Random

Algorithm:

If there are K arms, randomly select each arm with probability 1/K each step

Example below: Each arm gets on average 2,500 “pulls” out of 10,000

Pure exploration: We will probably learn the best arm *eventually*, but not very efficient

Strategy 1: Epsilon-Greedy

The Algorithm:

With prob ε, pick a random arm (explore). With prob 1−ε, pick best arm so far (exploit).

Simple to implement and understand

ε is a an important hyperparameter: too high = over-explore

Fixed ε = explore forever, even after convergence

Variant: decaying ε over time

Strategy 2: Upper Confidence Bound (UCB)

The Algorithm:

1) For each arm *i*, compute the **upper confidence bound**:

average reward so far

$$\bar{X}_i + \sqrt{\frac{2\ln(N)}{n_i}}$$

total pulls

pulls for arm i

2) Pull the arm with the highest bound.

3) Repeat.

"Optimism in the face of uncertainty"

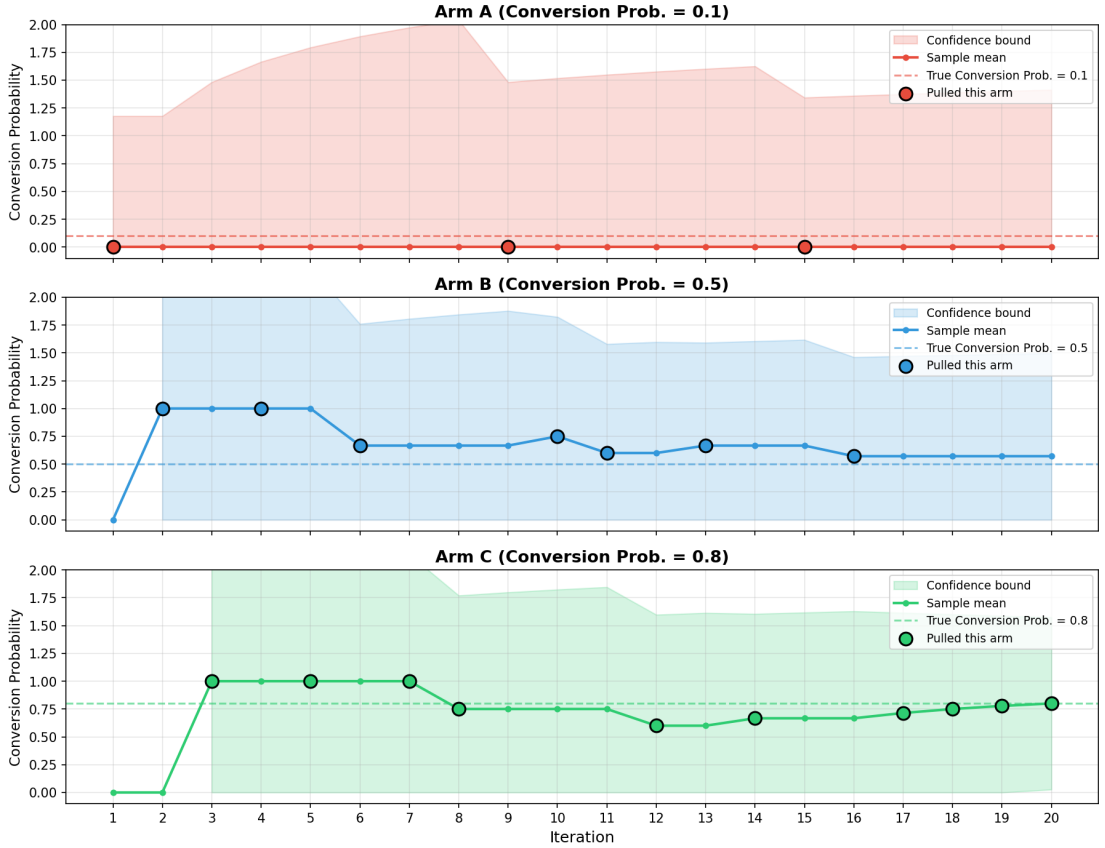
Frequently pulled arms → more certainty and lower ceiling

Rarely pulled arms → lots of uncertainty, worth exploring due to high ceiling / potential

Exploration naturally decays as all arms get pulled

Strategy 3: Upper Confidence Bound (UCB)

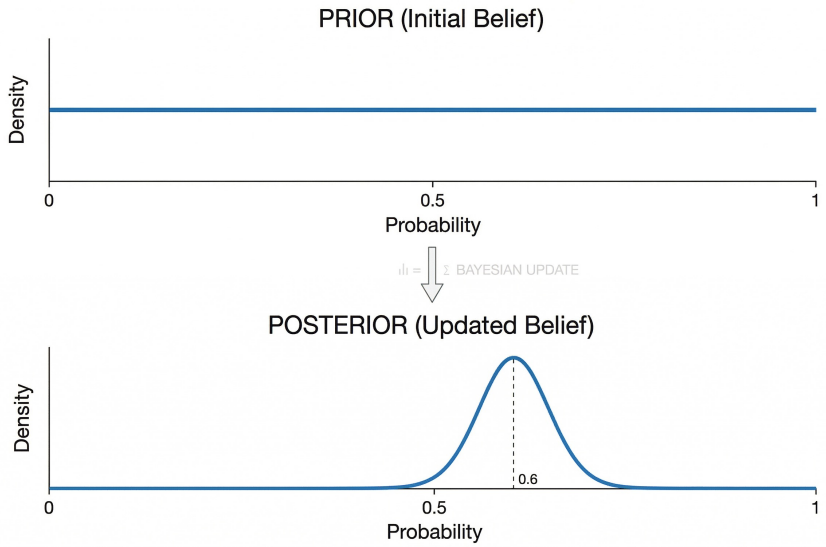
A simulated example:
Reward = 1 if successful conversion, 0 otherwise



Strategy 3: Thompson Sampling

Thompson sampling is the **gold standard** bandit algorithm in industry, used most widely

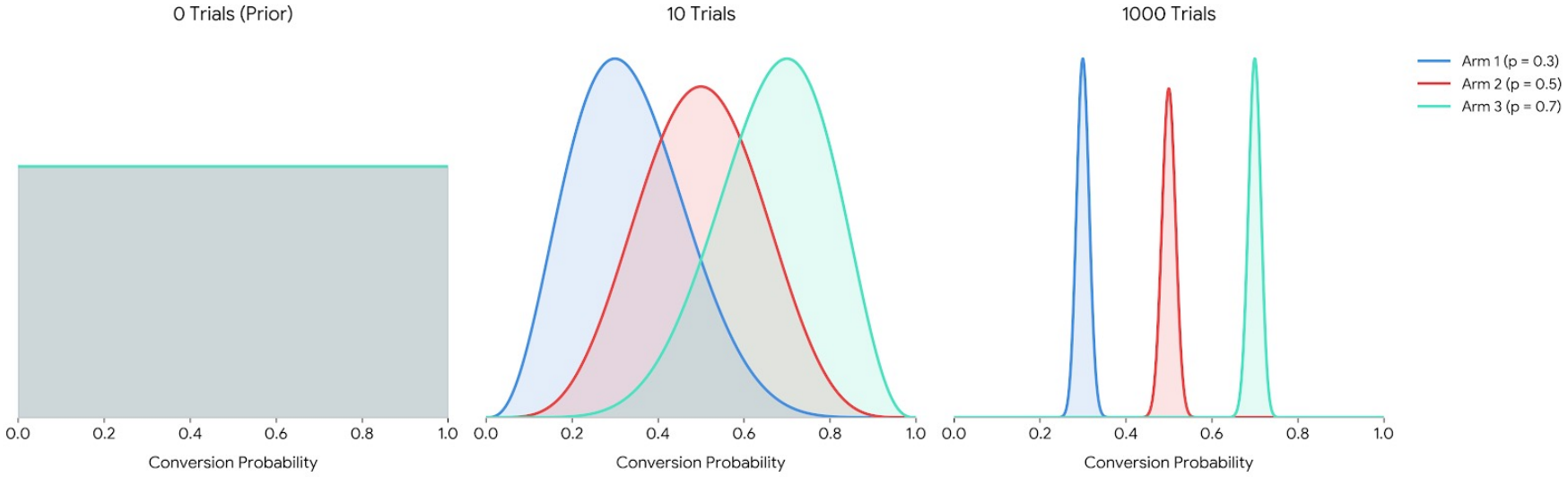
Based on Bayes' rule / Bayesian reasoning:



The Algorithm

1) Start with uniform prior for each arm
2) Sample from each arm's posterior
3) Pull arm with *highest sampled value*
4) Observe reward, update posterior, repeat

Strategy 3: Thompson Sampling



Summary of Bandit Algorithms

Epsilon-Greedy

Approach

Random arm with prob ε, best arm otherwise

Strengths

Simplest to implement and understand

Weaknesses

ε requires tuning; explores forever even after convergence

Upper Confidence Bound

Approach

Pick arm with highest optimistic estimate

Strengths

Exploration decays naturally

Weaknesses

Deterministic; can over-explore underperforming arms early on

Thompson Sampling

Approach

Sample from each arm's posterior; pull highest sample

Strengths

Best empirical performance; most used in industry

Weaknesses

Requires choosing a prior; harder to explain intuitively

Contextual Bandits

Plain bandit: arms treated as equivalent across users

Contextual bandit: uses what we know about the user before choosing an arm

Context = features about the current situation (user, time, device, location).

Netflix Artwork Personalization

Contextual bandits take the *specific* user's information ("context") into account

Profile Type	Score Image A	Score Image B
Comedy	5.7	6.3
Romance	7.2	6.5

Image A

Image B

Figure 3: Example of contextual image selection based on the type of profile. Comedy refers to a profile that mostly watches comedy titles. Similarly, Romance watches mostly romantic titles. The contextual bandit selects the image of Robin Williams, a famous comedian, for comedy-inclined profiles while selecting an image of a kissing couple for profiles more inclined towards romance.

https://netflixtechblog.com/artwork-personalization-c589f074ad76

Multi-Armed Bandit Simulation

15-min Break

Reinforcement Learning

Matching Supply and Demand at Uber

Uber must decide which driver-rider pair to match

Location of driver matters, and matching decision affects next location

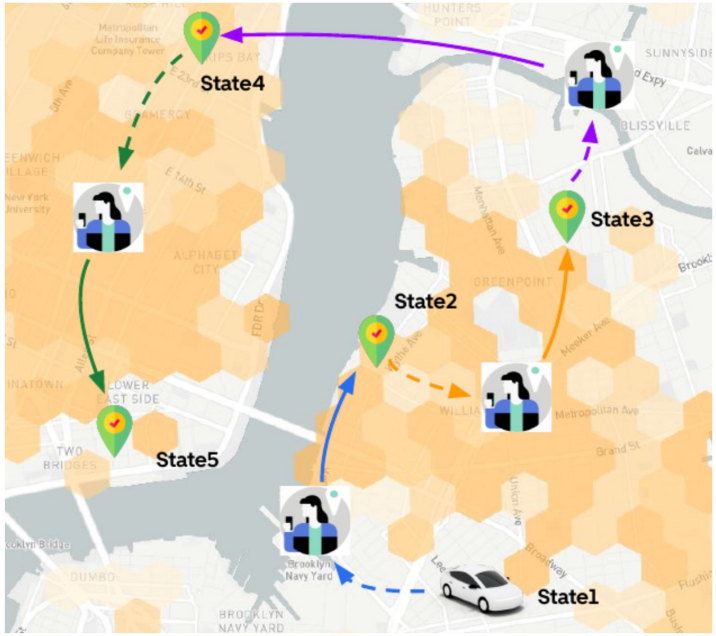


Figure 3: Trajectory of a driver through completing multiple trips.

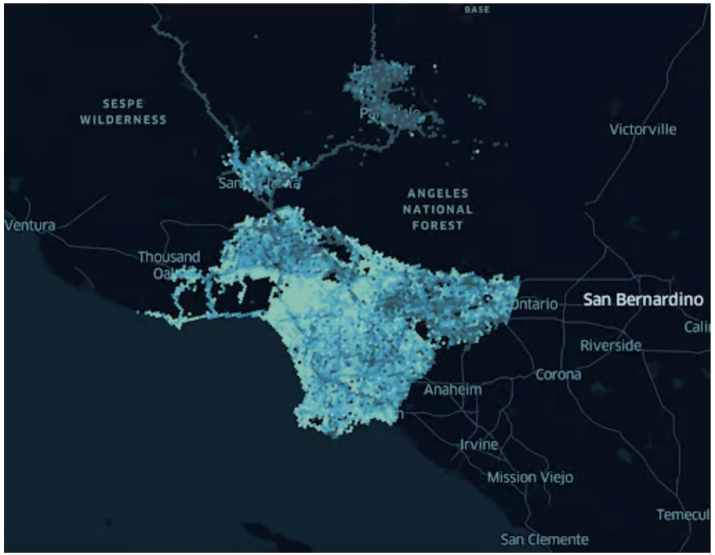


Figure 4: Example of target value function in Los Angeles

https://www.uber.com/blog/reinforcement-learning-for-modeling-marketplace-balance/

Reinforcement Learning in Practice

McKinsey Analytics

It's time for businesses to chart a course for reinforcement learning

An advanced artificial intelligence technique is quickly becoming accessible to organizations as a tool for speeding innovation and solving complex business problems.

by Jacomo Corbo, Oliver Fleming, and Nicolas Hohn

Exhibit

Initial applications of reinforcement learning span most, if not all, industries.

● Optimizing product development cycles (AI-assisted design) ● Optimizing complex operations ● Informing next best action for each customer

Industry	Sample reinforcement learning applications
Advanced electronics and semiconductors	● Optimize silicon and chip design to increase performance and reduce manufacturing costs ● Optimize fabrication manufacturing process for improved yield and throughput
Agriculture	● Solve scheduling and production allocation challenges to increase yield ● Optimize network and warehouse logistics for reduced waste and costs ● Apply advanced pricing and promotion to improve product margins
Aerospace and defense	● Optimize engineering design processes to reduce time to market for new systems and improve quality
Automotive	● Optimize design processes to shorten development cycle for new cars and features and improve quality ● Deploy advanced predictive maintenance to prevent rare failures and unplanned outages ● Deliver real-time production monitoring and controls to increase manufacturing yield
Financial services	● Apply real-time trading and pricing strategies for greater agility and revenue ● Optimize ATM replenishment and allocation strategies to reduce costs and improve the customer experience ● Deliver advanced personalization capabilities that adapt promotions, offers, and recommendations daily for increased customer satisfaction and sales

https://www.mckinsey.com/capabilities/quantumblack/our-insights/its-time-for-businesses-to-chart-a-course-for-reinforcement-learning

Mining	● Optimize design process so teams can explore a greater range of mine designs for improving mine yield ● Use intelligent process controls for managing power generation and bore milling to increase yield and reduce costs ● Apply holistic logistics scheduling to optimize mine-to-shipping operations and reduce costs
Oil and gas	● Enable real-time well monitoring and precision drilling for increased yield ● Optimize tanker routing to reduce costs and ensure on-time delivery ● Enable advanced predictive maintenance to prevent rare equipment failures and unplanned outages
Pharmaceuticals	● Optimize drug discovery, identifying molecules of interest faster to reduce the time and cost of research and bring new therapies to market faster ● Automate chemistry, manufacturing, and controls (CMC) to maximize batch yield and quality ● Optimize biological methods to reach peak production output
Retail	● Optimize routing, logistics network planning, and warehouse operations to reduce costs and keep shelves stocked ● Implement advanced inventory modeling and digitize supply-chain planning to prevent out-of-stocks and waste ● Deliver advanced personalization capabilities that adapt promotions, offers, and recommendations daily for increased customer satisfaction and sales
Telecom	● Optimize network layout to maximize coverage and minimize power consumption ● Manage networks in real time to optimize service quality and reduce downtime ● Apply advanced personalization to increase cross-sell and upsell revenue
Transport and logistics	● Optimize routing, logistics network planning, and warehouse operations to reduce costs and improve customer satisfaction ● Optimize inbound and outbound delivery networks to minimize shipping delays and associated costs

The Key Idea: States

The critical idea in reinforcement learning is the **state: A snapshot of the current situation**

Dynamic pricing: inventory left, day of the week

Ride-hailing: location of drivers, time of day

Robotics: Joint angles, velocity

Example: Selling Cakes

Each weekend, a bakery makes two specialty cakes. For each of Friday, Saturday, and Sunday, they must decide whether to sell at $15 or $35. Assume exactly one customer comes in each day.

	$15	$35
Friday	0.4	0.02
Saturday	0.5	0.15
Sunday	0.9	0.8

true, unknown purchase probabilities

What is the optimal (i.e., revenue maximizing) pricing policy?

The Reinforcement Learning Framework

State (s)	**Current situation** — e.g., (day of the week, number of cakes left)
Action (a)	**Agent's choice** — e.g., Price: $15 or $35
Reward (r)	**Immediate feedback** — e.g., Revenue from today's cake sales
Policy (π)	**Strategy** — states → actions — e.g., What price to set for each (day, inventory) combo
Value Q(s,a)	**Expected cumulative future reward** — Total future revenue if taking action a in state s

Q-Learning

Q-Learning

State space:

Time: Friday, Saturday, Sunday

Inventory: 2, 1, 0

6 states total

The **Q-value** tells us the expected total discounted reward from being in state s, taking action a, and then acting optimally from that point on.

"If I set price **[a]** on **[day]** with **[this much inventory],** *and* play optimally for the rest of the weekend, how much total revenue will I make from this point forward?"

The "Q-Table"

	$15	**$35**
(Friday, 2)	?	?
(Saturday, 2)	?	?
(Saturday, 1)	?	?
(Sunday, 2)	?	?
(Sunday, 1)	?	?
(Sunday, 0)	?	?

Q-Learning

Q-Learning update rule:

$$Q(s,a) \leftarrow (1-\alpha) \cdot \underbrace{Q(s,a)}_{\text{current best estimate}}$$

The "Q-Table"

	$15	$35
(Friday, 2)	?	?
(Saturday, 2)	?	?
(Saturday, 1)	?	?
(Sunday, 2)	?	?
(Sunday, 1)	?	?
(Sunday, 0)	?	?

Q-Learning

Q-Learning update rule:

$$Q(s,a) \leftarrow (1-\alpha) \cdot \underbrace{Q(s,a)}_{\text{current best estimate}} + \alpha \cdot \underbrace{\left[r + \gamma \cdot \max_{a'} Q(s',a')\right]}_{\text{estimate from this episode}}$$

α (learning rate) How much to update toward new info

γ (discount) How much to weight future vs. immediate reward

r Immediate reward received

max Q(s′,a′) Highest Q-value from new state (under best action)

The "Q-Table"

	$15	$35
(Friday, 2)	?	?
(Saturday, 2)	?	?
(Saturday, 1)	?	?
(Sunday, 2)	?	?
(Sunday, 1)	?	?
(Sunday, 0)	?	?

Q-Learning Convergence

	$15	$35
(Fri, 2)	26.7	14.6
(Sat, 2)	22.8	6.8
(Sun, 1)	14.6	4.3
(Sun, 2)	0	8.7
(Sun, 1)	7	14.7
(Sun, 0)	0	0

100 episodes

	$15	$35
(Fri, 2)	30	28.7
(Sat, 2)	26.6	23.4
(Sun, 1)	15.1	23.2
(Sun, 2)	11.1	17.5
(Sun, 1)	13.5	19.6
(Sun, 0)	0	0

500 episodes

(switched!)

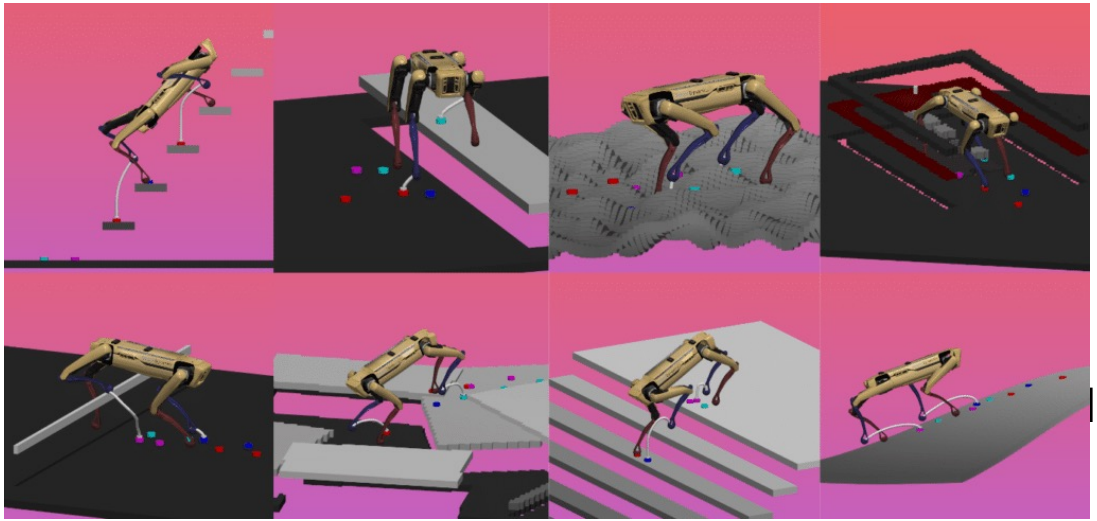
	$15	$35
(Fri, 2)	36.9	36.1
(Sat, 2)	30.9	25.8
(Sun, 1)	15.3	23.5
(Sun, 2)	11.5	19.2
(Sun, 1)	11.9	23.7
(Sun, 0)	0	0

1,000 episodes

Reinforcement learning is extremely data hungry – needs lots of episodes to learn the optimal policy correctly

Q-Learning Simulation

Reinforcement Learning in Robotics



https://bostondynamics.com/blog/starting-on-the-right-foot-with-reinforcement-learning/

@Intuitive-robots

Reinforcement Learning in Gaming

State: ~20K values: hero stats, items, map vision, cooldowns

Actions: ~170K options: move, attack, cast, buy

Reward: Win/loss + small bonuses for kills, gold, objectives

"OpenAI Five plays 180 years worth of games against itself every day, learning via self-play…running on 256 GPUs and 128,000 CPU cores"

https://openai.com/index/openai-five/

Lunar Lander Simulation (Offline)

Assignment 3

Glossary (1/2)

Reinforcement Learning	A family of methods where an agent learns to make good decisions through trial and error, receiving rewards for its actions.
Agent	The decision-maker in a reinforcement learning problem; the entity that chooses actions and learns from the rewards it receives.
Action	A choice the agent can make at each step, such as setting a price, placing a stone, or applying a motor torque.
Reward	The immediate numerical feedback the agent receives after taking an action, such as revenue from a sale.
Exploration vs. Exploitation	The core trade-off between trying new actions to gather information (exploration) and choosing the best-known action (exploitation).
Multi-Armed Bandit	A framework for choosing among options with unknown reward rates, balancing exploration and exploitation to maximize total reward.
Epsilon-Greedy	A simple bandit strategy: with probability epsilon pick a random option (explore), otherwise pick the current best (exploit).
Upper Confidence Bound (UCB)	A bandit strategy that picks the option with the highest optimistic estimate, giving a bonus to less-tried options.
Thompson Sampling	A Bayesian bandit strategy that samples from each option's probability distribution and picks whichever sample is highest.

Glossary (2/2)

Posterior Distribution	An updated probability distribution that reflects both prior beliefs and observed data, used in Thompson Sampling.
Contextual Bandit	A bandit that uses features about the current situation (user, time, device) to choose an arm, rather than treating every round identically.
State	A snapshot of the current situation in a reinforcement learning problem, such as the day of the week and inventory level.
Policy	A strategy mapping each state to an action; the goal of RL is to find the policy that maximizes total reward.
Q-Value	The expected total future reward from taking a specific action in a specific state and then acting optimally afterward.
Q-Learning	An algorithm that learns optimal Q-values through repeated trial and error, filling in a table of state-action values.
Discount Factor (gamma)	A number between 0 and 1 that controls how much the agent values future rewards relative to immediate ones.