ST455: Reinforcement Learning

Lecture 1: Introduction to Reinforcement Learning

Chengchun Shi

Lecture Outline

1. **Introduction and Course Overview**

2. **Multi-Armed Bandit**

3. **Contextual Bandits**

Lecture Outline

1. **Introduction and Course Overview**
2. Multi-Armed Bandit
3. Contextual Bandits

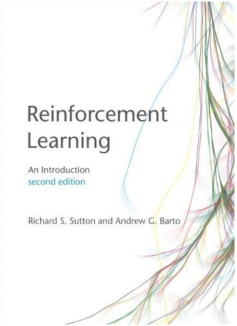
Course Information

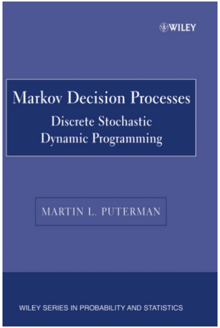
- **Lectures**: Tue 14 – 16:00 pm, MAR 2.04 (zoom: 985 785 4435)
- **Seminars**: Wed 13 – 14:30 pm, CBG 1.05 (zoom: 985 785 4435; lead by CS)
  Wed 16 – 17:30 pm, CBG 2.04 (lead by Edward)

  You may join lecture and seminar via **zoom** as well
- **Office Hours**:
  - Chengchun Shi (c.shi7@lse.ac.uk): Tue 13-14:00 pm, COL 8.08 or ZOOM
  - Edward Plumb (e.plumb@lse.ac.uk): Fri 11-12 am.
  - Please use **LSE Student Hub** to book slots
- **Assessment**:
  - Two summative assignments at Weeks 4 & 7 (10% each)
  - A final project (group project) to apply/develop RL algorithms (80%)
- We use **GitHub**. Please register and fill in the link
- More on **Moodle** (link)

Textbooks

- **Reinforcement Learning: An Introduction** (Second Edition) by Sutton and Barto (2018)
  - **Hardcover** £50 on Amazon
  - **Ebook** free online (link)
  - ≈70K citations so far
- **Markov decision processes: discrete stochastic dynamic programming** by Puterman (2014)





Useful Resources

- **Deepmind & UCL** reinforcement learning (RL) course by David Silver
  - **Course webpage** link
  - **Videos** available on Youtube
  - **Slides** available on webpage
- **UC Berkeley** PhD-level deep RL course by Sergey Levine
  - **Course webpage** link
  - Some more **resources** link
- Working draft on "**Reinforcement Learning: Theory and Algorithms**" by Alekh, Nan, Sham and Wen link

Applications

(a) Games

(b) Health Care

(c) Ridesharing

(d) Robotics

(e) Finance

(f) Automated Driving

Games

Figure: AlphaGo. See Silver et al. [2016] for details. To be discussed in more detail in Lecture 9.

Games (Cont'd)

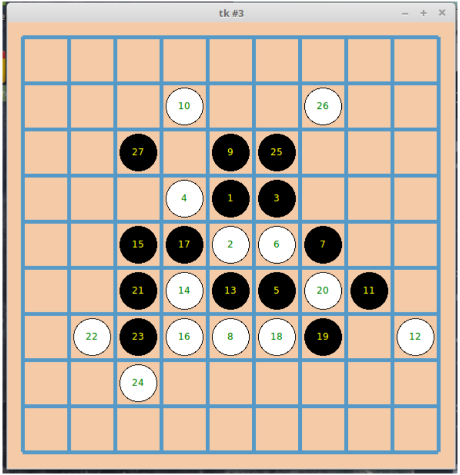
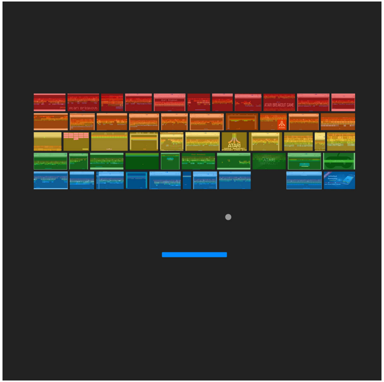


Figure: An implementation of AlphaGo Zero on Gomoku. To be discussed in more detail in Seminar 10.

Games (Cont'd)



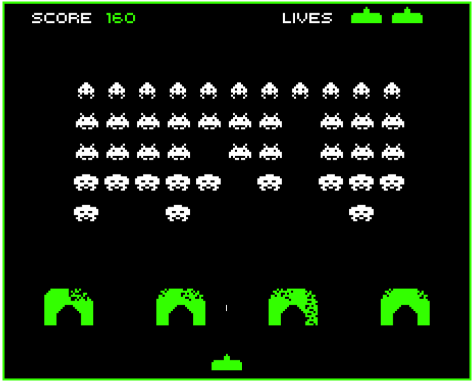


Figure: Two Atari Games: Breakout (link) and Space Invaders. To be discussed in more detail in Lecture 7 & Seminar 8.

Healthcare

- Management of **Type-I diabetes** [Luckett et al., 2019, Shi et al., 2020, 2022, Zhou et al., 2022a]
- **Subject**: Patients with Type-I diabetes
- **Objective**: Improve health outcomes
- **Intervention**: Determine whether a patient needs to **inject insulin or not** based on their glucose levels, food intake, exercise intensity, etc.
- **Data**: OhioT1DM dataset

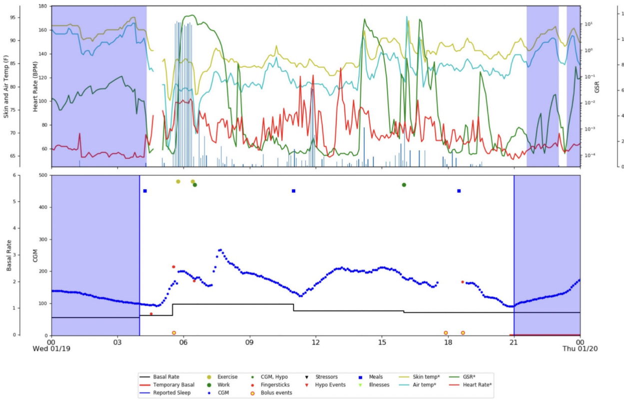


Figure: OhioT1DM data. To be discussed in Lecture 10.

Healthcare (Cont'd)

- **Intern health study** [NeCamp et al., 2020, Li et al., 2022]
- **Subject**: First-year medical interns working in stressful environments (e.g., long work hours and sleep deprivation)
- **Objective**: Promote physical and mental well-beings
- **Intervention**: Determine whether to send certain text message to a subject

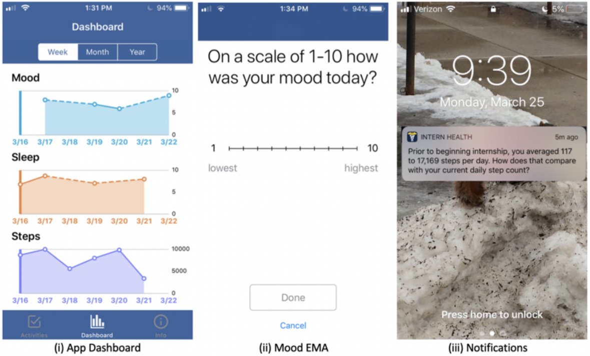


Figure: IHS. To be discussed in Lecture 10.

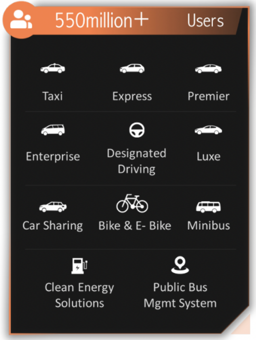
Healthcare (Cont'd)

Table 1. Examples of 6 different groups of notifications.

Notification groups	Life insight	Tip
Mood	Your mood has ranges from 7 to 9 over the past 2 weeks. The average intern's daily mood goes down by 7.5% after intern year begins.	Treat yourself to your favorite meal. You've earned it!
Activity	Prior to beginning internship, you averaged 117 to 17,169 steps per day. How does that compare with your current daily step count?	Exercising releases endorphins which may improve mood. Staying fit and healthy can help increase your energy level.
Sleep	The average nightly sleep duration for an intern is 6 hours 42 minutes. Your average since starting internship is 7 hours 47 minutes.	Try to get 6 to 8 hours of sleep each night if possible. Notice how even small increases in sleep may help you to function at peak capacity & better manage the stresses of internship.

- Other applications:
  - HeartSteps [Liao et al., 2020]
  - Sepsis treatment [Li et al., 2020, Chen et al., 2022, Zhou et al., 2022b]

Ridesharing



106TB+ vehicle trajectory data/day

4875TB+ data processed/day

40billion+ routing requests/day

15billion+ location points/day

Ridesharing (Cont'd)

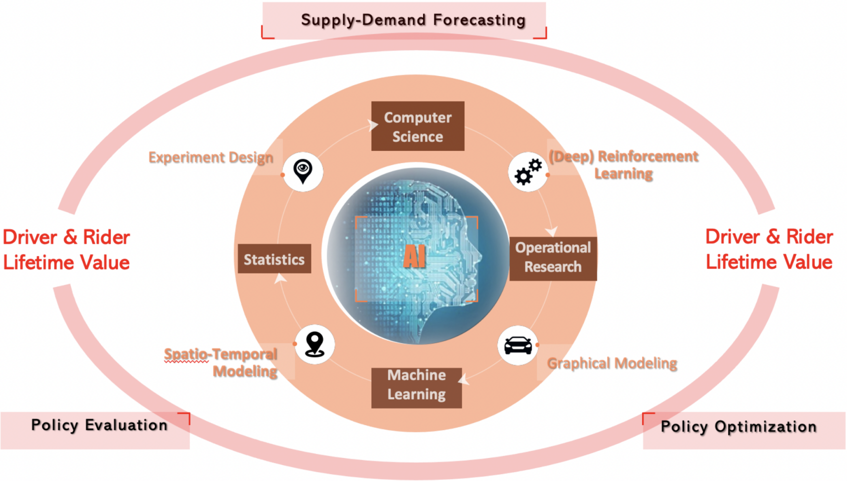


Figure: RL for Ridesharing. To be discussed in more detail in Lecture 7 & Seminar 7.

Robotics

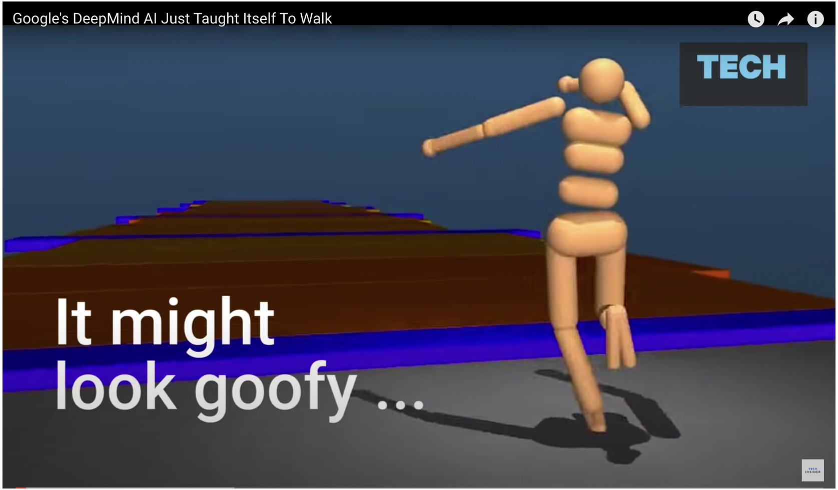


Figure: See https://www.youtube.com/watch?v=gn4nRCC9TwQ

ChatGPT

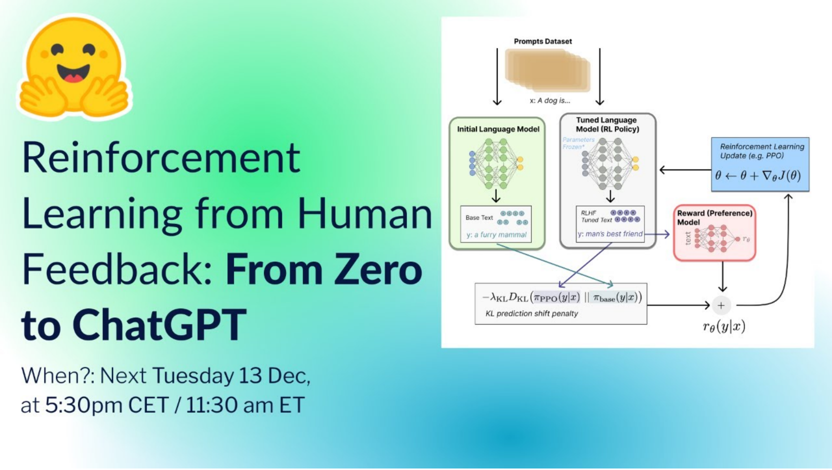
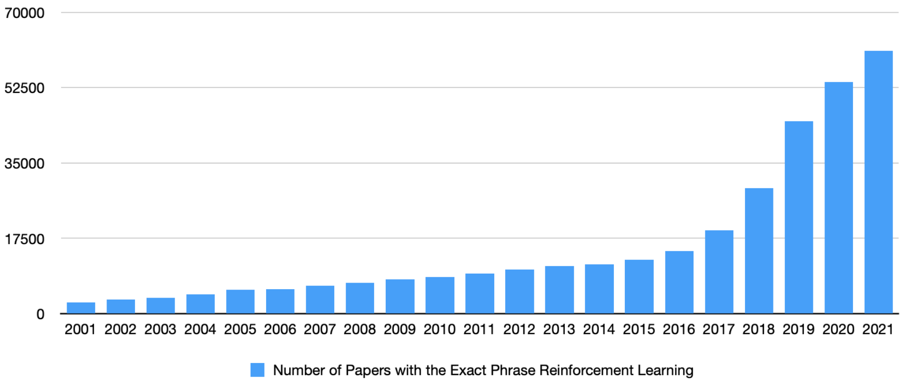


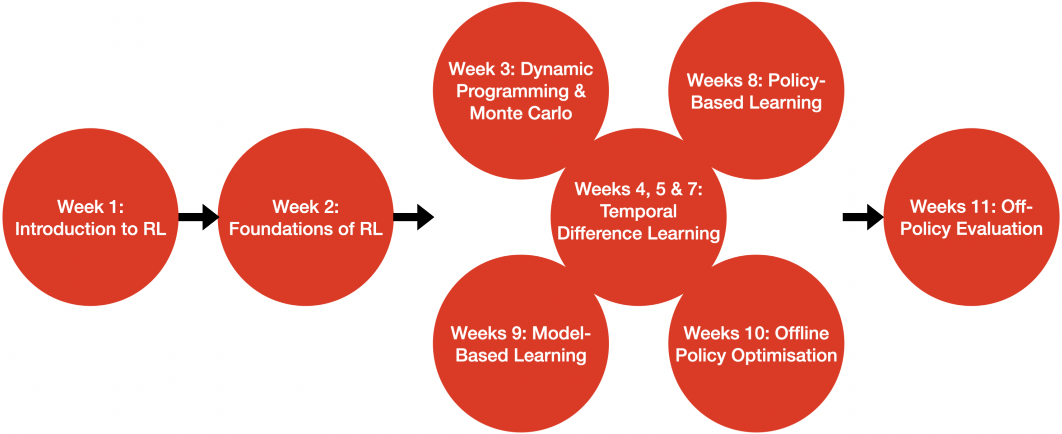
Figure: See https://www.youtube.com/watch?v=2MBJOuVq380

RL as a Research Topic

- One of the most vibrant research topics in **machine learning**
- Over 100 papers accepted at **ICML** 2020, accounting for more than 10% in total

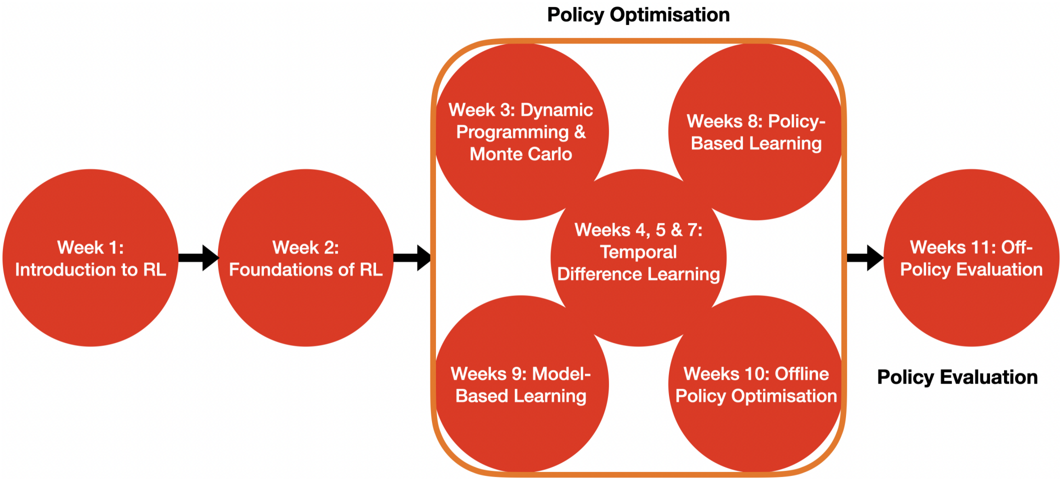


Roadmap



Week 1: Introduction to RL
Week 2: Foundations of RL
Week 3: Dynamic Programming & Monte Carlo
Weeks 8: Policy-Based Learning
Weeks 4, 5 & 7: Temporal Difference Learning
Weeks 9: Model-Based Learning
Weeks 10: Offline Policy Optimisation
Weeks 11: Off-Policy Evaluation

Roadmap (Cont'd)



1. Introduction and Course Overview

2. Multi-Armed Bandit

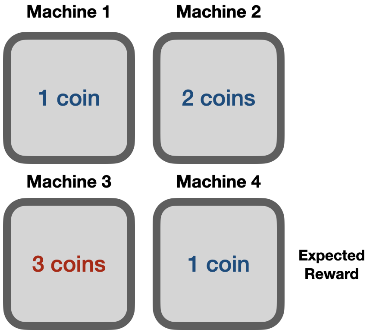
3. Contextual Bandits

Multi-Armed Bandit (MAB) Problem

- The **simplest** RL problem
- A casino with **multiple** slot machines
- Playing each machine yields an independent **reward**.
- Limited knowledge (unknown reward distribution for each machine) and resources (**time**)
- **Objective**: determine which machine to pick at each time to maximize the expected **cumulative rewards**

Multi-Armed Bandit Problem (Cont'd)

- k-armed bandit problem (k machines)
- $A_t \in \{1, \cdots, k\}$: arm (machine) pulled (experimented) at time t
- $R_t \in \mathbb{R}$: reward at time t
- $Q(a) = \mathbb{E}(R_t | A_t = a)$ expected reward for each arm a (**unknown**)
- **Objective**: maximize $\sum_{t=1}^{T} \mathbb{E} R_t$.



Greedy Action Selection

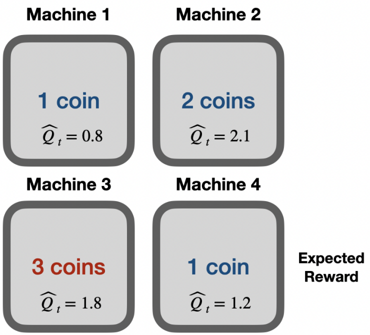
- **Action-value methods**: estimate the expected reward (i.e., value) of actions and use these estimates to select actions
- Estimated reward at time t:

$$\widehat{Q}_t(a) = \frac{\sum_{i=1}^{t} R_i \mathbb{I}(A_i = a)}{\sum_{i=1}^{t} \mathbb{I}(A_i = a)}$$

- **Greedy policy**:

$$A_t = \arg\max_a \widehat{Q}_{t-1}(a).$$

- Might be **suboptimal** in the long run.



Exploration-Exploitation Dilemma

- **Exploitation**: To maximize reward, the agent prefers the greedy policy that selects actions that maximizes the estimated expected reward.
- **Exploration**: To discover which actions yield a higher reward, the agent must try actions that it has less selected to improve the estimation accuracy.
- **Trade-off** between exploration and exploitation:
  - Neither exploration nor exploitation can be used exclusively.
  - The agent must try various actions and progressively favour high-reward actions.
- Practical algorithms: ε-**greedy**, **upper confidence bound** (UCB), **Thompson sampling**.

ε-Greedy

- **Input**: Choose a small value parameter $\boldsymbol{\varepsilon} \in (0, 1)$.
- At each step **perform**:
  - With probability $\mathbf{1} - \varepsilon$: adopt the **greedy policy**;
  - With probability ε: choose a **randomly selected arm** from the set of all arms.
- Combines exploration and exploitation:
  - At each time, each arm is selected with probability at least $\boldsymbol{k}^{-1}\varepsilon$.
  - Greedy action is selected with probability $\mathbf{1} - \varepsilon + \boldsymbol{k}^{-1}\varepsilon$.

Incremental Implementation

- Average reward received from arm a by time t:

$$\widehat{Q}_t(a) = \mathbb{N}_t^{-1}(a) \sum_{i=1}^{t} \mathbb{I}(A_i = a) R_i,$$

  where $\mathbb{N}_t(a) = \sum_{i=1}^{t} \mathbb{I}(A_i = a)$.
- If arm a is selected at time $t+1$, then

$$\begin{aligned}
\widehat{Q}_{t+1}(a) &= \{\mathbb{N}_t(a) + 1\}^{-1} \left\{ \sum_{i=1}^{t} \mathbb{I}(A_i = a) R_i + R_{t+1} \right\} \\
&= \frac{\mathbb{N}_t(a)}{\mathbb{N}_t(a) + 1} \left\{ \mathbb{N}_t^{-1}(a) \sum_{i=1}^{t} \mathbb{I}(A_i = a) R_i \right\} + \frac{R_{t+1}}{\mathbb{N}_t(a) + 1} \\
&= \frac{\mathbb{N}_t(a)}{\mathbb{N}_t(a) + 1} \widehat{Q}_t(a) + \frac{R_{t+1}}{\mathbb{N}_t(a) + 1}.
\end{aligned}$$

Algorithm

- **Input**: $0 < \varepsilon < 1$, termination time $\boldsymbol{T}$.
- **Initialization**: $\boldsymbol{t} = \mathbf{0}$, $\widehat{\boldsymbol{Q}}(\boldsymbol{a}) = \mathbf{0}$, $\mathbb{N}(\boldsymbol{a}) = \mathbf{0}$, for $\boldsymbol{a} = \mathbf{1}, \mathbf{2}, \cdots, \boldsymbol{k}$.
- **While $\boldsymbol{t} < \boldsymbol{T}$**:
  - **Update $\boldsymbol{t}$**: $\boldsymbol{t} \leftarrow \boldsymbol{t} + \mathbf{1}$.
  - **ε-greedy action selection**:

$$\boldsymbol{a}^* \leftarrow \begin{cases} \arg\max_{\boldsymbol{a}} \widehat{\boldsymbol{Q}}(\boldsymbol{a}), & \text{with probabiltiy } \mathbf{1} - \varepsilon, \\ \text{random arm}, & \text{with probabiltiy } \varepsilon. \end{cases}$$

  - **Receive reward $\boldsymbol{R}$** from arm $\boldsymbol{a}^*$.
  - **Update** $\mathbb{N}(\boldsymbol{a}^*)$: $\mathbb{N}(\boldsymbol{a}^*) \leftarrow \mathbb{N}(\boldsymbol{a}^*) + \mathbf{1}$.
  - **Update** $\widehat{\boldsymbol{Q}}(\boldsymbol{a}^*)$:

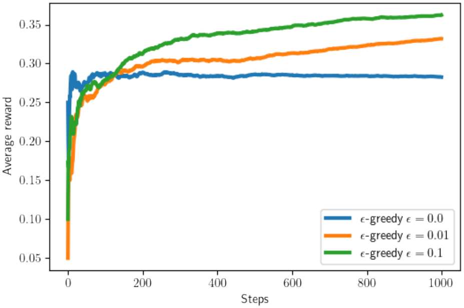
$$\widehat{\boldsymbol{Q}}(\boldsymbol{a}^*) \leftarrow \frac{\mathbb{N}(\boldsymbol{a}^*) - \mathbf{1}}{\mathbb{N}(\boldsymbol{a}^*)} \widehat{\boldsymbol{Q}}(\boldsymbol{a}^*) + \frac{\mathbf{1}}{\mathbb{N}(\boldsymbol{a}^*)} \boldsymbol{R}.$$

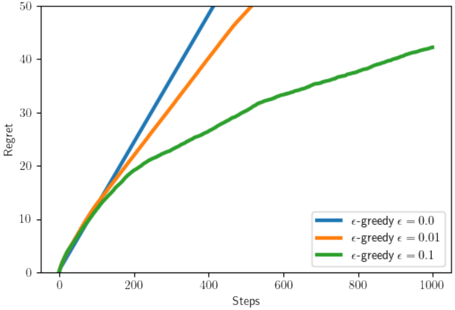
Example: Four Bernoulli Arms

Reward distributions	Bernoulli(0.1)	Bernoulli(0.4)	Bernoulli(0.1)	Bernoulli(0.1)

Best arm

Example: Four Bernoulli Arms (Cont'd)





Tracking Nonstationarity

- Incremental update:

$$\widehat{Q}(a^*) \leftarrow \frac{\mathbb{N}(a^*) - \mathbf{1}}{\mathbb{N}(a^*)} \widehat{Q}(a^*) + \frac{\mathbf{1}}{\mathbb{N}(a^*)} R.$$

- Alternatively, for a given step size parameter $0 < \alpha < 1$,

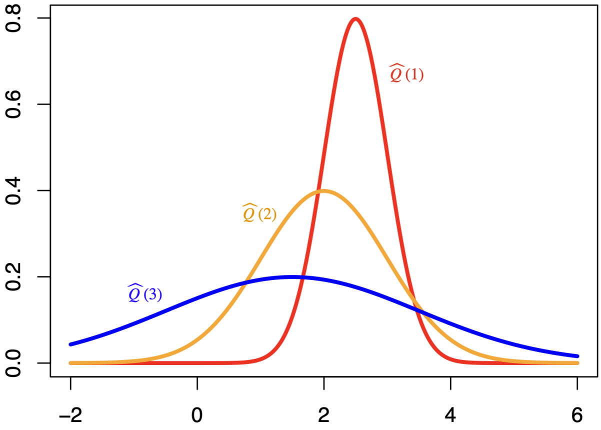
$$\widehat{Q}(a^*) \leftarrow (\mathbf{1} - \alpha)\widehat{Q}(a^*) + \alpha R.$$

- Give more **weight** to recently observed reward. Handles **nonstationarity** (reward distribution varies over time).
- **Exponential weighted moving average**:

$$\begin{aligned}\widehat{Q}(a^*) \leftarrow \alpha R + (\mathbf{1} - \alpha)\widehat{Q}^{(-1)}(a^*) &\leftarrow \alpha R + \alpha(\mathbf{1} - \alpha) R^{(-1)} + (\mathbf{1} - \alpha)^2 \widehat{Q}^{(-2)}(a^*) \\ &\leftarrow \alpha R + \alpha \sum_{i=1}^{J} (1 - \alpha)^i R^{(-i)}.\end{aligned}$$

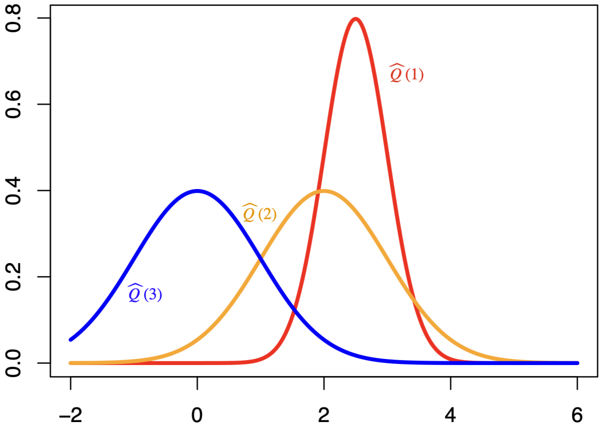
Optimism in the Face of Uncertainty

- The **optimistic principle**:
- The more **uncertain** we are about an action-value;
- The more **important** it is to explore that action;
- It could be the **best** action.
- Likely to pick blue action.
- **Different** from ε-greedy which selects arms uniformly random.



Optimism in the Face of Uncertainty (Cont'd)

- After picking blue action;
- Become less **uncertain** about the value;
- More likely to pick other actions;
- Until we home in on best action.



Upper Confidence Bound

- Estimate an **upper confidence** $U_t(a)$ for each action value such that

$$Q(a) \leq \widehat{Q}_t(a) + U_t(a),$$

  with high probability.
- $U_t(a)$ quantifies the **uncertainty** and depends on $\mathbb{N}_t(a)$ (number of times arm a has been selected up to time t)
  - Large $\mathbb{N}_t(a) \rightarrow$ small $U_t(a)$;
  - Small $\mathbb{N}_t(a) \rightarrow$ large $U_t(a)$.
- Select actions maximizing upper confidence bound

$$a^* = \arg\max_a [\widehat{Q}_t(a) + U_t(a)].$$

- Combines **exploration** ($U_t(a)$) and **exploitation** ($\widehat{Q}_t(a)$).

Upper Confidence Bound (Cont'd)

- Set $U_t(a) = \sqrt{c \log(t)/\mathbb{N}_t(a)}$ for some positive constant c.
- According to **Hoeffding's inequality** (link), when rewards are bounded between 0 and 1, the event

$$Q(a) \leq \widehat{Q}_t(a) + U_t(a),$$

holds with probability at least $1 - t^{-2c}$ (converges to 1 as $t \to \infty$).

Algorithm

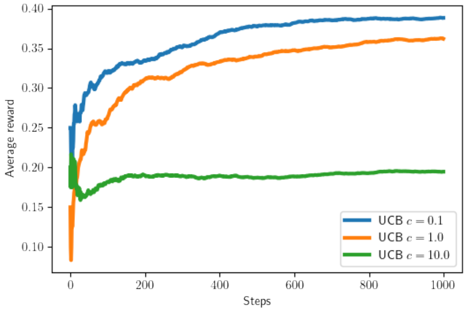
- **Input**: some positive constant c, termination time T.
- **Initialization**: $t = 0$, $\widehat{Q}(a) = 0$, $\mathbb{N}(a) = 0$, for $a = 1, 2, \cdots, k$.
- **While** $t < T$:
  - **Update** t: $t \leftarrow t + 1$.
  - **UCB action selection**:

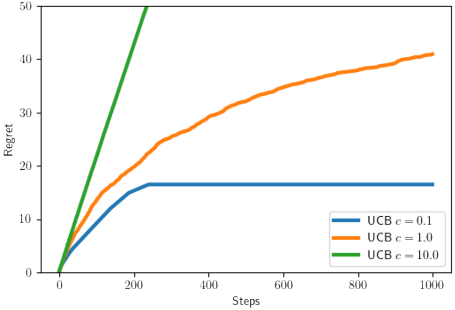
$$a^* \leftarrow \arg\max_a [\widehat{Q}(a) + \sqrt{c \log(t)/\mathbb{N}_t(a)}].$$

  - **Receive reward** R from arm a^*.
  - **Update** $\mathbb{N}(a^*)$: $\mathbb{N}(a^*) \leftarrow \mathbb{N}(a^*) + 1$.
  - **Update** $\widehat{Q}(a^*)$:

$$\widehat{Q}(a^*) \leftarrow \frac{\mathbb{N}(a^*) - 1}{\mathbb{N}(a^*)} \widehat{Q}(a^*) + \frac{1}{\mathbb{N}(a^*)} R.$$

Example: Four Bernoulli Arms (Revisited)





Thompson Sampling

- A **highly-competitive** algorithm to address exploration-exploitation trade-off.
- Impose **statistical models** for the reward distribution with parameter θ.
- Impose **prior distributions** for θ.
- At time t,
  - Use **Bayes rule** to update the **posterior distribution** of θ.
  - Sample a model parameter θ_t from the posterior distribution.
  - Compute action-value given θ_t, i.e., $\mathbb{E}(R|A = a, \theta_t)$.
  - Select action maximizing action-value

$$a^* = \arg\max_a \mathbb{E}(R|A = a, \theta_t).$$

- Posterior distribution quantifies the **uncertainty** of the estimated model parameter (**exploration**).
- $\mathbb{E}(R|A = a, \theta_t)$ estimates the oracle action value (**exploitation**).

Thompson Sampling (Cont'd)

- **Statistical models**:
  - $p(r|a, \theta)$ models the probability density/mass function of rewards under arm a.
  - $p(\theta)$ models the probability density/mass function of θ.
- **Bayesian inference**:
  - Likelihood function $\ell_t(\theta) = \prod_{i=1}^{t} p(R_i|A_i, \theta)$.
  - Compute the posterior distribution according to Bayes rule

$$p_t(\theta|\mathcal{D}) = \frac{p(\theta)\ell_t(\theta)}{\int_\theta p(\theta)\ell_t(\theta)d\theta} \propto p(\theta)\ell_t(\theta),$$

  where $\mathcal{D}$ denotes the observed data.

- **Compute action value**:

$$\mathbb{E}(R|A = a, \theta_t) = \int_r rp(r|a, \theta_t)dr.$$

Thompson Sampling (Bernoulli Bandit Example)

- **Statistical models**:
  - Reward of the ath arm follows a Bernoulli distribution with mean $\theta(a)$.
  - $\theta(a)$ follows a Beta(α, β) distribution (**prior**).
  - Why Beta distribution?
    - Commonly used distribution for outcomes bounded between $\mathbf{0}$ and $\mathbf{1}$
    - Reduced to **uniform** distribution when $\alpha = \beta = 1$
    - **Conjugate** distribution of binomial, i.e. posterior distribution is Beta as well
    - α and β measures the beliefs for **success** and **failure**
- **Bayesian inference**:
  - $\theta(a)$ follows a Beta$(S_a + \alpha, F_a + \beta)$ distribution (**posterior**) where (S_a, F_a) corresponds to the success and failure counters under arm a.
- **Compute action value**:

$$\mathbb{E}(R|A = a, \theta_t) = \theta_t(a).$$

Algorithm (Bernoulli Bandit Example[1])

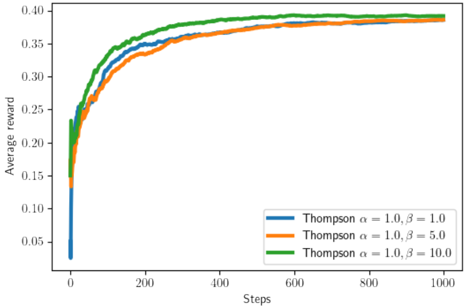
- **Input**: hyper-parameters α, $\beta > 0$, termination time T.
- **Initialization**: $t = 0$, $S_a = F_a = 0$, for $a = 1, 2, \cdots, k$.
- **While** $t < T$:
  - **Update** t: $t \leftarrow t + 1$.
  - **Posterior sampling**: For $a = 1, 2, \cdots, k$, sample

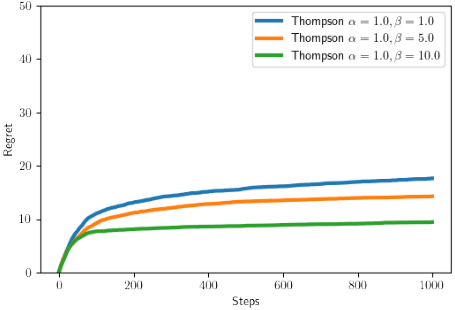
$$\theta_a \sim \mathrm{Beta}(S_a + \alpha, F_a + \beta)$$

  - **Action selection**: $a^* \leftarrow \arg\max_a \theta_a$.
  - **Receive reward** R from arm a^*.
  - **Update** S_a and F_a:
    - If $R = 1$, $S_a \leftarrow S_a + 1$;
    - If $R = 0$, $F_a \leftarrow F_a + 1$.

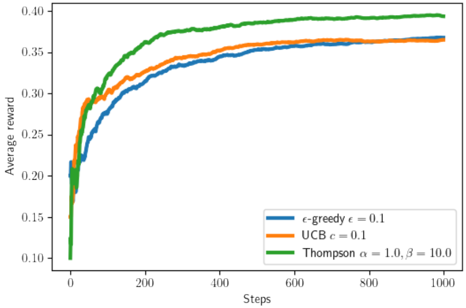
[1] The general algorithm can be found in Chapelle and Li [2011]

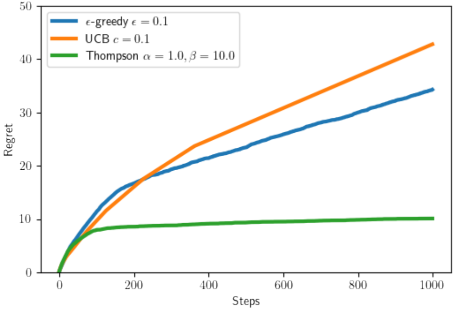
Example: Four Bernoulli Arms (Revisited)





Example: Four Bernoulli Arms (Cont'd)





Theory

Define the **regret** T, $\mathcal{R}(T)$ as the difference between the cumulative reward under the **best action** and that under the **selected actions**, up to time T.

Theorem (UCB, Auer et al. [2002])

*The **expected regret** of the UCB algorithm* $\mathbb{E}\mathcal{R}(T)$ *is upper bounded by* $C_1 \log(T)$ *for some constant* $C_1 > 0$.

Theorem (TS, Agrawal and Goyal [2012])

*The **expected regret** of the Thompson sampling algorithm* $\mathbb{E}\mathcal{R}(T)$ *is upper bounded by* $C_2 \log(T)$ *for some constant* $C_2 > 0$.

- Both algorithms achieve logarithmic expected regret.
- Their performances are nearly the same as the oracle method that works as if the best action were known.
- ε-Greedy algorithm with a constant ε has a **linear** expected regret (proportional to T). More to discuss in seminar class.

1. Introduction and Course Overview

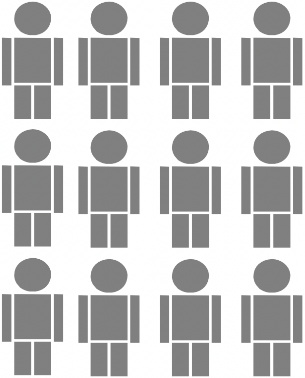
2. Multi-Armed Bandit

3. Contextual Bandits

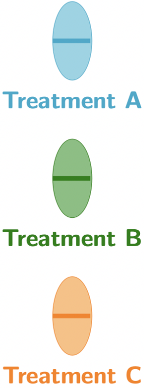
Contextual Bandits

- Extension of MAB with **contextual** information.
- A **widely-used** model in medicine and technological industries.
- At time $\boldsymbol{t}$, the agent
  - Observe a context $\boldsymbol{S_t}$;
  - Select an action $\boldsymbol{A_t}$;
  - Receives a reward $\boldsymbol{R_t}$ (depends on both $\boldsymbol{S_t}$ and $\boldsymbol{A_t}$).
- **Objective**: maximize cumulative reward.
- ε**-greedy**, **UCB** and **Thompson sampling** can be similarly adopted [see e.g., Chu et al., 2011, Agrawal and Goyal, 2013, Zhou et al., 2020, Zhang et al., 2020].

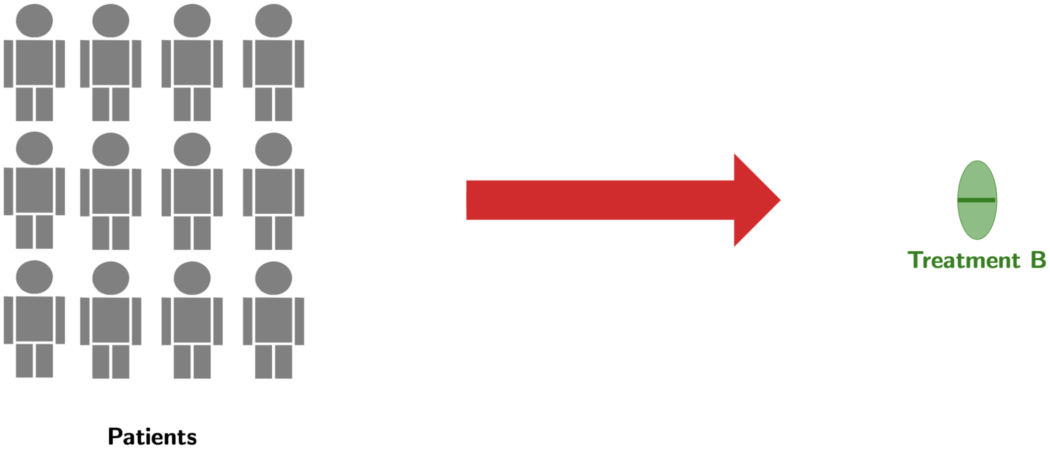
Application I: Precision Medicine



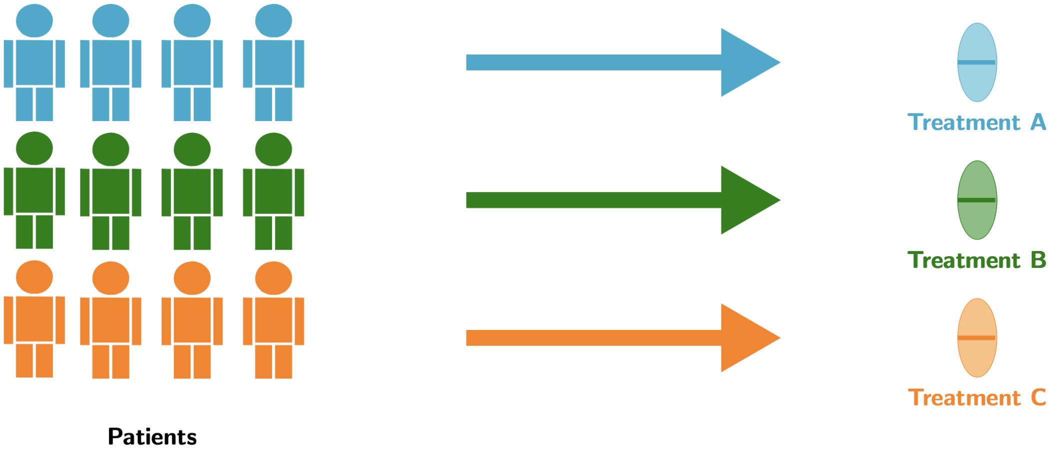
Patients



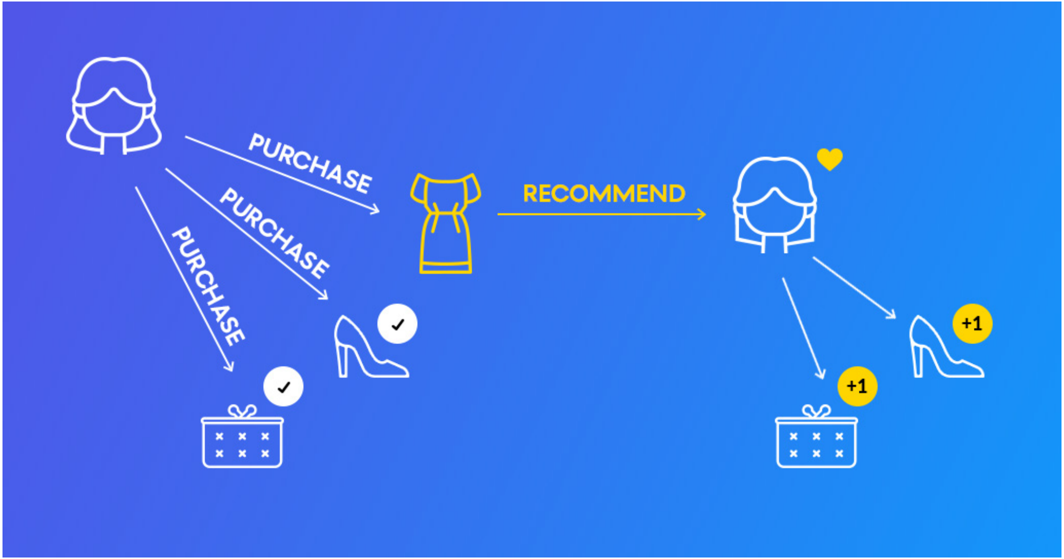
One-Size-Fits-All



Individualized Treatment Regime



Application II: Personalized Recommendation



Contextual Bandits Applications

- S_t: Patient's or customer's baseline characteristics
- A_t: Treatment (product) recommended to the patient (customer)
- R_t: Patient's outcome or customer's action

Summary

- Exploration-exploitation trade-off
- ε-greedy, UCB (the optimistic principle) and Thompson sampling
- Multi-armed bandits, contextual bandits

Seminar Exercises

- Get started with **OpenAI Gym** (link)
- Multi-armed bandits problem

References I

Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on learning theory*, pages 39–1. JMLR Workshop and Conference Proceedings, 2012.

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, pages 127–135. PMLR, 2013.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. *Advances in neural information processing systems*, 24:2249–2257, 2011.

Elynn Y Chen, Rui Song, and Michael I Jordan. Reinforcement learning with heterogeneous data: Estimation and inference. *arXiv preprint arXiv:2202.00088*, 2022.

References II

Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 208–214. JMLR Workshop and Conference Proceedings, 2011.

Luchen Li, Ignacio Albert-Smet, and Aldo A Faisal. Optimizing medical treatment for sepsis in intensive care: from reinforcement learning to pre-trial evaluation. *arXiv preprint arXiv:2003.06474*, 2020.

Mengbing Li, Chengchun Shi, Zhenke Wu, and Piotr Fryzlewicz. Testing stationarity and change point detection in reinforcement learning. *arXiv preprint arXiv:2203.01707*, 2022.

References III

Peng Liao, Kristjan Greenewald, Predrag Klasnja, and Susan Murphy. Personalized heartsteps: A reinforcement learning algorithm for optimizing physical activity. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(1):1–22, 2020.

Daniel J Luckett, Eric B Laber, Anna R Kahkoska, David M Maahs, Elizabeth Mayer-Davis, and Michael R Kosorok. Estimating dynamic treatment regimes in mobile health using v-learning. *Journal of the American Statistical Association*, 2019.

Timothy NeCamp, Srijan Sen, Elena Frank, Maureen A Walton, Edward L Ionides, Yu Fang, Ambuj Tewari, and Zhenke Wu. Assessing real-time moderation for developing adaptive mobile health interventions for medical interns: Micro-randomized trial. *Journal of medical Internet research*, 22(3):e15033, 2020.

References IV

Chengchun Shi, Runzhe Wan, Rui Song, Wenbin Lu, and Ling Leng. Does the markov decision process fit the data: Testing for the markov property in sequential decision making. pages 8807–8817, 2020.

Chengchun Shi, Sheng Zhang, Wenbin Lu, and Rui Song. Statistical inference of the value function for reinforcement learning in infinite-horizon settings. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 84(3):765–793, 2022.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Weitong Zhang, Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural thompson sampling. *arXiv preprint arXiv:2010.00827*, 2020.

References V

Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural contextual bandits with ucb-based exploration. In *International Conference on Machine Learning*, pages 11492–11502. PMLR, 2020.

Wenzhuo Zhou, Ruoqing Zhu, and Annie Qu. Estimating optimal infinite horizon dynamic treatment regimes via pt-learning. *Journal of the American Statistical Association*, 0 (0):1–14, 2022a.

Yunzhe Zhou, Zhengling Qi, Chengchun Shi, and Lexin Li. Optimizing pessimism in dynamic treatment regimes: A bayesian learning approach. *arXiv preprint arXiv:2210.14420*, 2022b.

Questions