Multi-armed bandits

Exploration-Exploitation Dilemma

Cathy Wu

6.7920: Reinforcement Learning: Foundations and Methods

Readings

1. Aleksandrs Slivkins. Introduction to Multi-Armed Bandits. 2019. Chapters 1, 8.

Outline

1. From RL to bandits
2. Exploration Strategies
3. Linear and contextual linear bandits

Outline

1. **From RL to bandits**
   a. Example: Recommender systems
   b. Regret
2. Exploration Strategies
3. Linear and contextual linear bandits

From RL to Multi-armed Bandit

for $i = 1, \dots, n$ **do**

~~1. Set $t = 0$~~

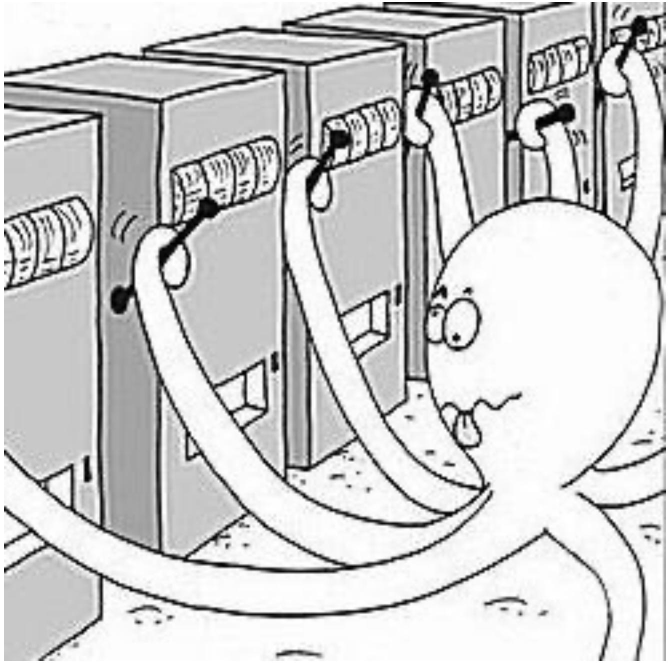
~~2. Set initial state s_0~~

~~3. **while** (s_t not terminal)~~

1) Take action a_t
2) Observe ~~next state s_{t+1} and~~ reward r_t

~~**endwhile**~~

endfor



From RL to Multi-armed Bandit

The *protocol*

for $i = 1, \ldots, n$ **do**

1. Take action a_t
2. Observe reward $r_t \sim v(a_t)$

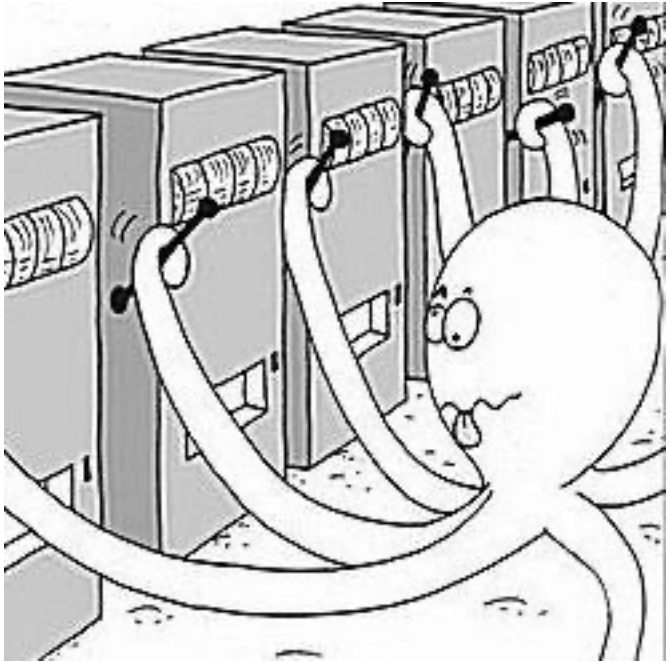
endfor

The *problem*

- Set of A actions
- Reward distribution $v(a)$ with $\mu(a) = \mathbb{E}[r(a)]$ (bounded in [0,1] for convenience)

The *objective*

- Maximize sum of reward $\mathbb{E}[\sum_{t=1}^{n} r_t]$

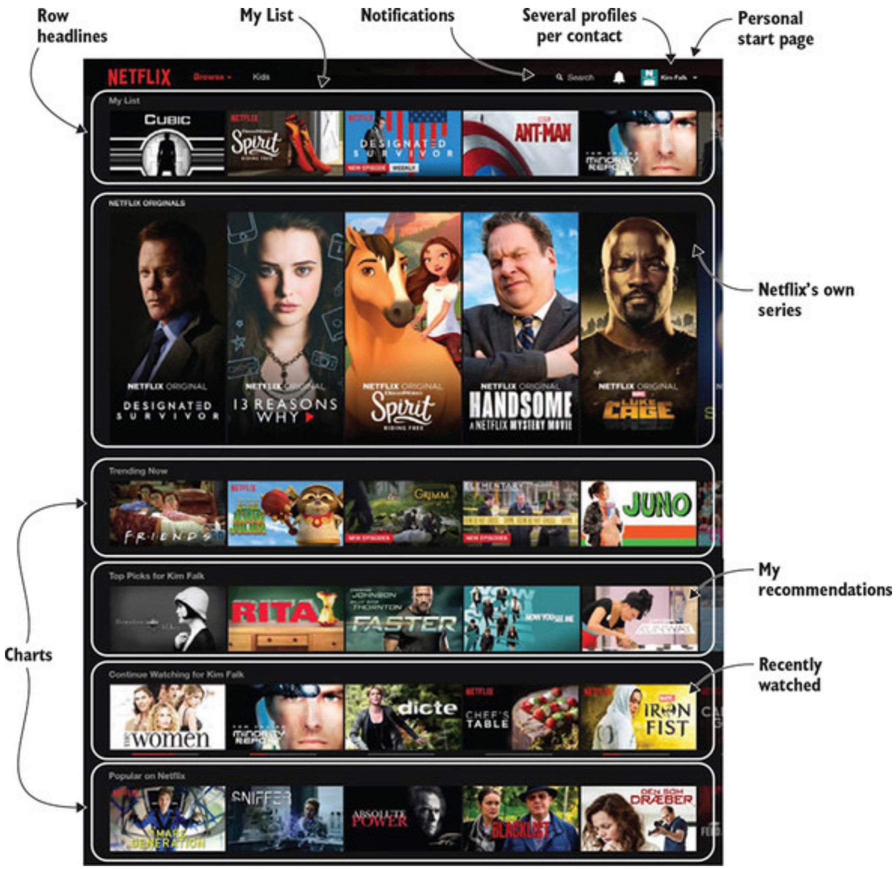


Why study bandits?

- Bandits simplify the RL interaction loop (MDP), providing a focused problem setting to consider the role of exploration in sequential decision making (exploration-exploitation dilemma).
- Also called online learning, methods for analyzing bandits are foundational for finite sample analysis in RL – that is, convergence rate, as opposed to asymptotic convergence.
- Contextual bandits are the most widely deployed form of RL, in the form of recommender systems. Understanding bandits means understanding the core ideas and algorithms behind these products and services.

A Simple Recommendation System

- A RS can recommend different genres of movies (e.g. action, adventure, romance, animation)
- Users arrive at random and no information about the user is available
- The RS picks a genre to recommend to the user but not the specific movies
- The feedback is whether the user watched a movie of the recommended genre or not
- Objective: Design a RS that maximizes the movies watched in the recommended genre



RS as a Multi-armed Bandit

for $i = 1, \ldots, n$ **do**

1. User arrives
2. Recommend genre a_t
3. Reward

$$r_t = \begin{cases} 1 & \text{user watches movie of genre } a_t \\ 0 & \text{otherwise} \end{cases}$$

endfor

RS as a Multi-armed Bandit

The *model*

- $v(a)$ is a Bernoulli
- $\mu(a) = \mathbb{E}[r(a)]$ is the probability a random user watches a movie of genre a
- Assumption: $r_t \sim v(a_t)$ is a realization of the Bernoulli of a genre a

The *objective*

- Maximize sum of reward $\mathbb{E}[\sum_{t=1}^{n} r_t]$

Other Examples

- Movies, TV, music
- Packet routing
- Clinical trials
- Web advertising
- Health advice
- Education
- Computer games
- Resource mining
- …

HeartSteps

HeartSteps explores new ways that mobile technology—smartphones and wearable activity trackers—can be used to help patients to increase their physical activity. We have developed a mobile app that works with a Fitbit activity tracker to help individuals set activity goals, plan how they will be active, and remain motivated to find ways to incorporate physical activity into their daily lives. Our ultimate goal is to develop technology that effectively supports physical activity over the long-term.

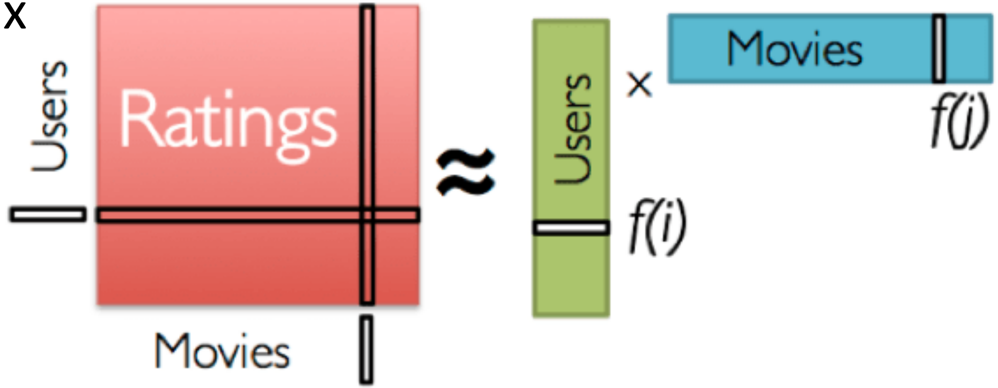
Funded by the National Institutes of Health (NIH)

https://heartsteps.net

Recommender system strategies: in a nutshell

- Studied since the 90s.
- Content-based filtering (early systems): Recommend items with features like the user's past selections.
- Collaborative filtering (modern systems): Recommend items based on other users with similar selection characteristics to the user's past selections.
  - Dominant approach: matrix factorization
  - Fueled by Netflix Prize competition (2006--): 100mil movie ratings

$$A = U\Sigma V^T = (U\sqrt{\Sigma})(\sqrt{\Sigma}V^T)$$



Fundamental challenges for recommender systems

- The cold-start problem: need selection data to base recommendations
  - How to recommend new items?
  - Example: new posts on social media, new webpages, new videos on YouTube
  - Many recommender systems are highly dynamic.
- Balancing short-term vs long-term optimization
- *Further reading (Bridging Systems):* Emerging ideas around designing recommender systems that "bridge" people rather than polarize them (https://bridging.systems/)

The exploration-exploitation dilemma

Problem 1: The environment does not reveal the reward of the actions not selected by the learner

- The learner should gain information by repeatedly selecting all actions $\Longrightarrow$ **exploration**

Problem 2: Whenever the learner selects a bad action, it suffers some regret

- The learner should reduce the regret by repeatedly selecting the best action $\Longrightarrow$ **exploitation**

Challenge: The learner should solve the exploration-exploitation dilemma!

Regret – formalizing the exploration-exploitation dilemma

$$R_n = \max_a \mathbb{E}\left[\sum_{t=1}^{n} r_t(a)\right] - \mathbb{E}\left[\sum_{t=1}^{n} r_t(a_t)\right]$$

The expectation summarizes any possible source of randomness (either in r or in the algorithm)

Relation to RL: Can think of this as n trajectories (of length 1).

Measures not only the final error, but all mistakes made over n "iterations."

The Regret

- Number of times action a has been selected after n rounds

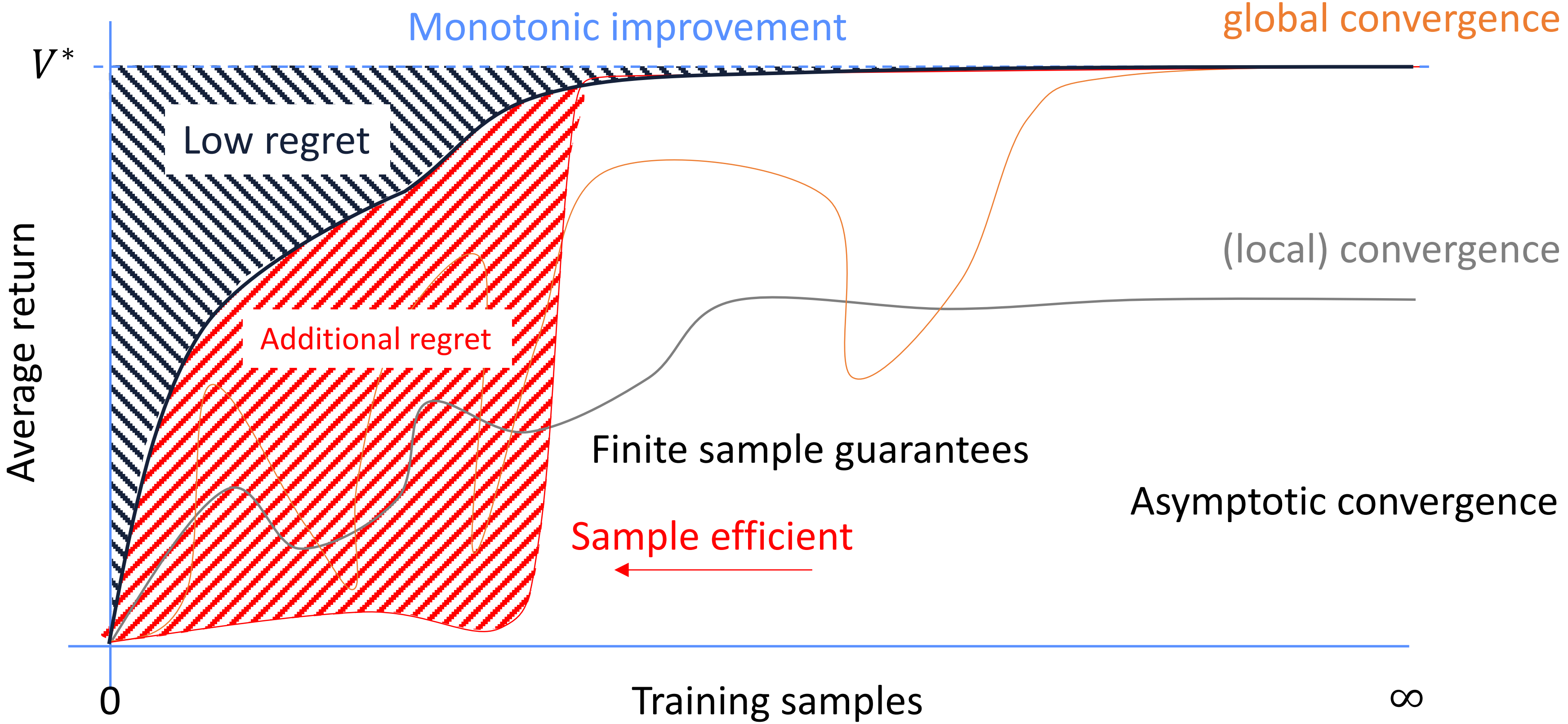
$$T_n(a) = \sum_{t=1}^{n} \mathbb{I}\{a_t = a\}$$

- Gap $\Delta(a) \coloneqq \mu(a^*) - \mu(a)$

$$R_n = \sum_{a \neq a^*} \mathbb{E}[T_n(a)]\Delta(a)$$

- We only need to study the expected number of times suboptimal actions are selected
- Worst case possible: $R_n = \mathcal{O}(n)$
  - **Discuss**: Why?
- A good algorithm has $R_n = o(n)$, i.e. $\frac{R_n}{n} \to 0$

What does it mean for an algorithm to work?

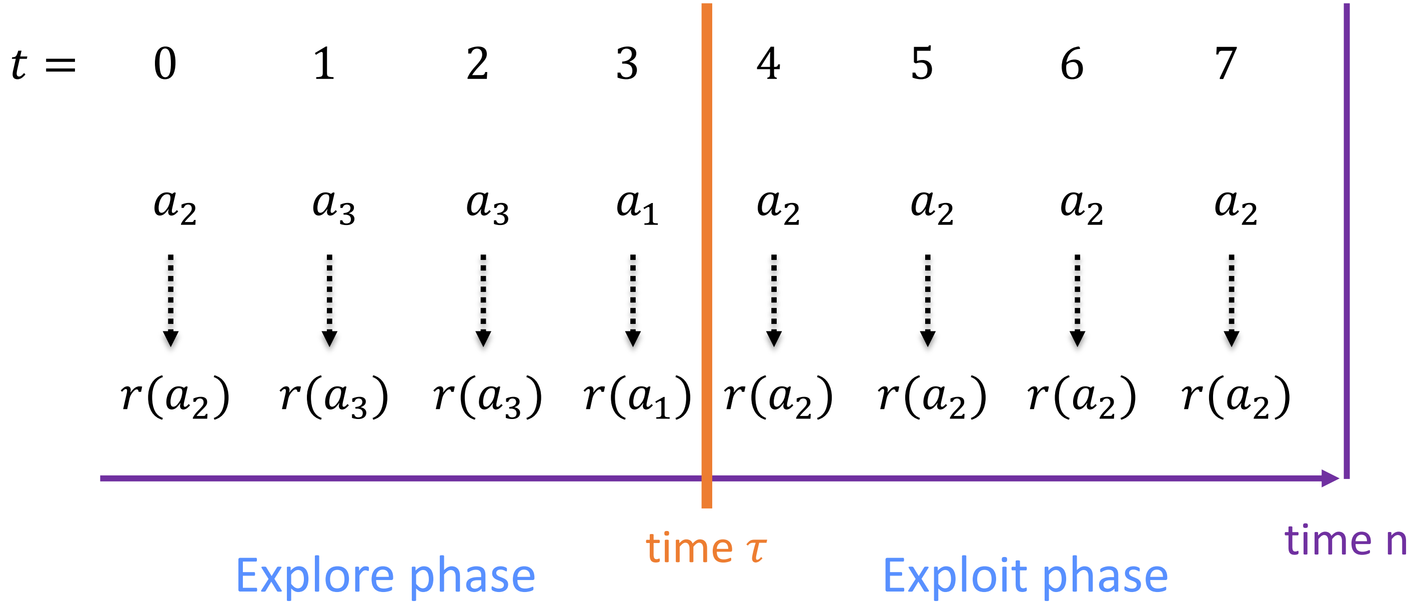


Outline

1. From RL to bandits
2. **Exploration Strategies**
    a. Explore-then-Commit
    b. ϵ-greedy
    c. Optimism in the face of uncertainty: upper confidence bound (UCB)
3. Linear and contextual linear bandits

Explore-then-Commit

Consider: $\mathcal{A} = \{a_1, a_2, a_3\}$; $A = |\mathcal{A}| = 3$



Explore-then-Commit: Algorithm

Define:

$$T_n(a) = \sum_{t=1}^{n} \mathbb{I}\{a_t = a\}$$

Explore phase

for $i = 1, \dots, \tau = AK$ **do**

1. Take action $a_t \sim \mathcal{U}(A)$ (or round robin)
2. Observe reward $r_t \sim v(a_t)$

endfor

Compute statistics for each action a

$$\hat{\mu}_\tau(a) = \frac{1}{T_\tau(a)} \sum_{s=1}^{\tau} r_s \mathbb{I}\{a_s = a\}$$

Exploit phase

for $i = \tau + 1, \dots, n$ **do**

1. Take action $\hat{a}^* = \arg\max_a \hat{\mu}_\tau(a)$
2. Observe reward $r_t \sim v(\hat{a}^*)$

endfor

Explore-then-Commit: Regret

Theorem

Let A be the number of arms. If explore-then-commit is run for n steps, exploring (round robin) for the first τ steps, then it suffers (expected) regret:

$$R_n \leq \tau + \mathcal{O}\left(\sqrt{\frac{A \log n}{\tau}} n\right)$$

- With best choice of τ, can get $R_n = \tilde{\mathcal{O}}\left(n^{\frac{2}{3}}\right)$
  - For $\tau = n^{2/3}(\log n)^{1/3}$
- Recall: worst possible: $R_n = \mathcal{O}(n)$
- HW5: You'll show that a tighter bound of $R_n = \mathcal{O}(\sqrt{n})$ is possible *when the gap $\Delta(a)$ is known*

$\tilde{\mathcal{O}}(\cdot)$ hides log factors

Concentration inequalities

- Foundational tools for regret analysis.

Proposition (Hoeffding Inequality)

Let $X_i \in [a, b]$ be an independent r.v. with common mean $\mu = \mathbb{E}X_i$.
Then:

$$\mathbb{P}[\underbrace{|\bar{X}_n - \mu|}_{\text{deviation}} > \underbrace{\epsilon}_{\text{accuracy}}] \leq \underbrace{2\exp\left(-\frac{2n\epsilon^2}{(b-a)^2}\right)}_{\text{confidence}} \quad \forall n > 0$$

where $\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$.

Explore-then-Commit: Regret Analysis

- Expected regret decomposition = explore phase + exploit phase

$$R_n = \sum_{t=1}^{\tau} \mathbb{E}[\nu(a^*) - \nu(a_t)] + \sum_{t=\tau+1}^{n} \mathbb{E}[\nu(a^*) - \nu(\hat{a}^*)] \le \tau + \sum_{t=\tau+1}^{n} \mathbb{E}[\nu(a^*) - \nu(\hat{a}^*)]$$

Recall:

$$\hat{\mu}_\tau(a) = \frac{1}{T_\tau(a)} \sum_{s=1}^{\tau} r_s \mathbb{I}\{a_s = a\}$$
$$\hat{a}^* = \arg\max_a \hat{\mu}_\tau(a)$$
$$\tau = KA$$

For exploit phase

- Define confidence radius $r(a) = \sqrt{\frac{2 \log n}{K}}$.
- Using Hoeffding's inequality, we get
$$\mathbb{P}[|\hat{\mu}_K(a) - \mu(a)| \le r(a)] \ge 1 - 2/n^4$$
- "Clean event" (above holds). Regret incurred when $\hat{a}^* \ne a^*$.
$$\mu(\hat{a}^*) + r(\hat{a}^*) \ge \hat{\mu}_K(\hat{a}^*) > \hat{\mu}_K(a^*) \ge \mu(a^*) - r(a^*)$$
$$\mu(a^*) - \mu(\hat{a}^*) \le r(\hat{a}^*) + r(a^*) = \mathcal{O}\left(\sqrt{\frac{\log n}{K}}\right)$$
- "Dirty event" (above doesn't hold). W.h.p., regret is bounded by $(n - \tau)2/n^4 \le \mathcal{O}(1/n^3)$. Small (can be neglected).

Overall expected regret:

$$R_n \le \tau + \mathcal{O}\left(\sqrt{\frac{\log n}{K}}(n - \tau)\right)$$
$$\le \tau + \mathcal{O}\left(\sqrt{\frac{\log n}{K}} n\right)$$

- Set $K = n^{2/3} A^{-2/3} (\log n)^{1/3}$, so that two sides are roughly equal. Get
$$R_n \le \mathcal{O}\left(n^{2/3} (A \log n)^{1/3}\right)$$

ϵ-greedy: Algorithm

[Recall: Q-learning]

for $i = 1, \ldots, n$ **do**

1. Take action
$$a_t = \begin{cases} \mathcal{U}(A) & \text{with probability } \epsilon_t \text{ (explore)} \\ \arg\max_a \hat{\mu}_t(a) & \text{with probability } 1 - \epsilon_t \text{ (exploit)} \end{cases}$$
2. Observe reward $r_t \sim v(a_t)$
3. Update statistics for action a_t
$$T_t(a_t) = T_{t-1}(a_t) + 1$$
$$\hat{\mu}_t(a_t) = \frac{1}{T_t(a_t)} \sum_{s=1}^{t} r_s \mathbb{I}\{a_s = a_t\}$$

endfor

ϵ-greedy: Regret

Theorem

If ϵ-greedy is run with parameter $\epsilon_t = t^{-\frac{1}{3}}(A \log t)^{1/3}$, then for each round t it suffers a regret:

$$R_t \leq \tilde{O}\left(t^{2/3}\right)$$

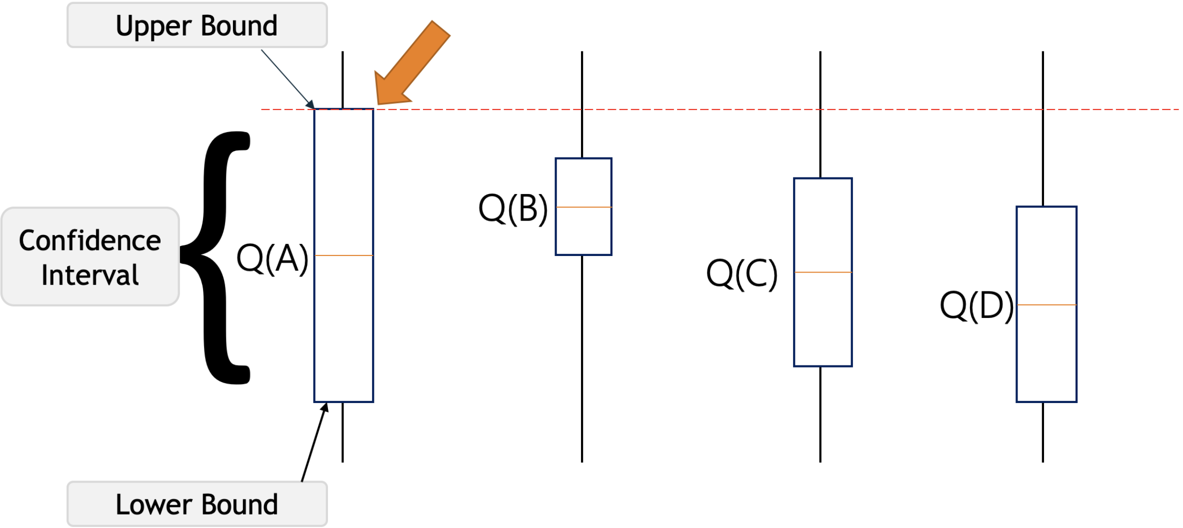
- Same asymptotic regret, now holds for all rounds t
- Can do better, but optimal ϵ depends on knowledge of Δ (difficult to tune) – same with explore-then-commit
- Keep selecting very bad arms with some probability
- Sharply separates exploration and exploitation

Types of exploration strategies

- Non-adaptive exploration
  - Explore-then-commit: explore + exploit (separately)
  - ϵ-greedy: exploit + explore (agnostic to exploitation)
- Adaptive exploration
  - Exploit + Explore (based on exploitation)

Optimism in Face of Uncertainty

"Whenever the value of an action is uncertain, consider its largest plausible value, and then select the best action."



Missing ingredient: uncertainty of our estimates.

Concentration inequalities

Proposition (Chernoff-Hoeffding Inequality)

Let $X_i \in [a, b]$ be n independent r.v. with mean $\mu = \mathbb{E}X_i$. Then:

$$\mathbb{P}\left[\left|\frac{1}{n}\sum_{t=1}^{n} X_t - \mu\right| > (b-a)\sqrt{\frac{\log\frac{2}{\delta}}{2n}}\right] \leq \delta$$

- Equivalent to Hoeffding's inequality
- Intuition: for a fixed probability, the estimated means will concentrate in a radius that shrinks with $\sqrt{n}$.

Recipe of UCB

1. Computation of estimates

$$\hat{\mu}_t(a) = \frac{1}{T_t(a)} \sum_{s=1}^{t} r_s \mathbb{I}\{a_s = a\}$$

2. Evaluation of uncertainty

$$|\hat{\mu}_t(a) - \mu(a)| \leq \sqrt{\frac{\log \frac{2}{\delta}}{2T_t(a)}}$$

3. Optimism: combine estimates and uncertainty (a.k.a. exploration bonus)

$$B_t(a) = \hat{\mu}_t(a) + \rho \sqrt{\frac{\log \frac{2}{\delta_t}}{2T_t(a)}}$$

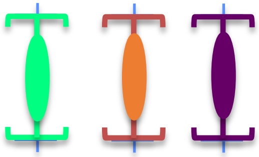
4. Select the best action (according to its combined value)

$$a_t = \arg\max_a B_t(a)$$

Upper Confidence Bound (UCB) Algorithm

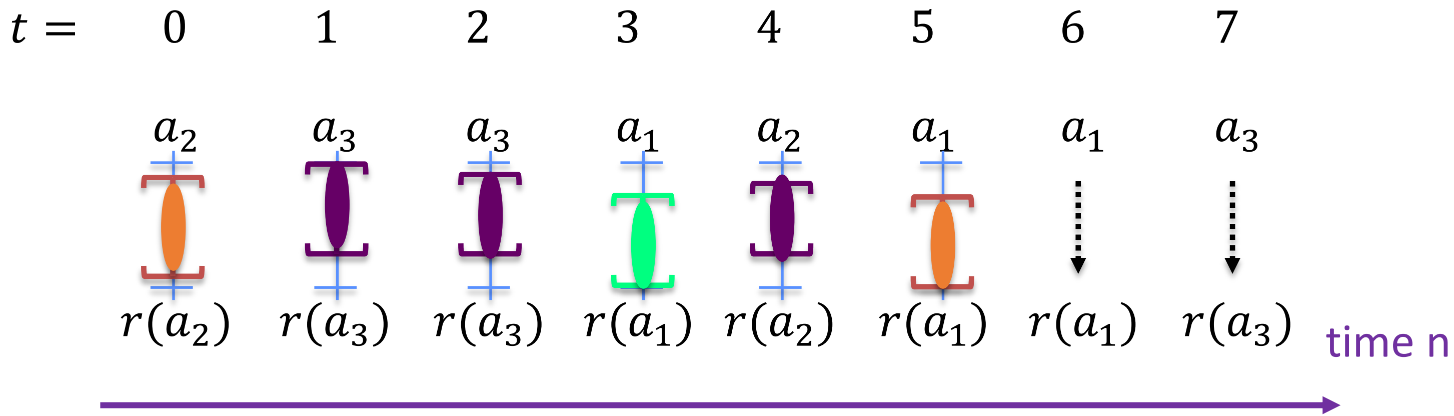
- Consider: $\mathcal{A} = \{a_1, a_2, a_3\}; \quad A = |\mathcal{A}| = 3$

Initial confidence intervals:



$a_1 \quad a_2 \quad a_3$

- $$a_t = \arg\max_i \underbrace{\hat{\mu}_t(a)}_{\text{exploitation}} + \underbrace{\rho \sqrt{\frac{\log \frac{2}{\delta_t}}{2T_t(a)}}}_{\text{exploration (bonus)}}$$



UCB: Algorithm

for $t = 1, \dots, n$ **do**

1. Compute upper-confidence bound
$$B_t(a) = \hat{\mu}_t(a) + \rho \sqrt{\frac{\log \frac{2}{\delta_t}}{2T_t(a)}}$$
2. Take action $a_t \arg\max_a B_t(a)$
3. Observe reward $r_t \sim v(a_t)$
4. Update statistics for action a_t
$$T_t(a_t) = T_{t-1}(a_t) + 1$$
$$\hat{\mu}_t(a_t) = \frac{1}{T_t(a_t)} \sum_{s=1}^{t} r_s \mathbb{I}\{a_s = a_t\}$$

endfor

UCB: Regret

Theorem

Consider a MAB problem with A Bernoulli arms with gaps $\Delta(a)$. If UCB is run with $\rho = 1$ and $\delta_t = \frac{1}{t}$ for n steps, then it suffers regret:

$$R_n = \mathcal{O}\left(\sum_{a \neq a^*} \frac{\log(n)}{\Delta(a)}\right)$$

- Can do better than non-adaptive exploration (explore-than-commit, ϵ-greedy)
- It (almost) matches lower bounds
- Does not require prior knowledge about the MAB, apart from range of the r.v.
- The big-O hides a few numerical constants and n-independent additive terms

UCB: Proof Sketch

- Disclaimer: This is a slightly suboptimal proof, but it provides a simpler proof strategy.
- Define the (high-probability) event *[statistics]*

$$\mathcal{E} = \left\{ \forall a, t \; |\hat{\mu}_t(a) - \mu(a)| \le \sqrt{\frac{\log \frac{2}{\delta}}{2T_t(a)}} \right\}$$

- By Chernoff-Hoeffding & union bound: $\mathbb{P}[\mathcal{E}] \ge 1 - nA\delta$
- If at time t, we select action a, then *[algorithm]*

$$B_t(a) \ge B_t(a^*)$$

$$\hat{\mu}_t(a) + \sqrt{\frac{\log \frac{2}{\delta}}{2T_t(a)}} \ge \hat{\mu}_t(a^*) + \sqrt{\frac{\log \frac{2}{\delta}}{2T_t(a^*)}}$$

- On the event $\mathcal{E}$, we have *[math]*

$$\mu(a) + 2\sqrt{\frac{\log \frac{2}{\delta}}{2T_t(a)}} \ge \mu(a^*)$$

UCB: Proof Sketch

- Assume t is the last time a is selected, then $T_n(a) = T_{t-1}(a) + 1$ (for $n \geq t$), thus:

$$\mu(a) + 2\sqrt{\frac{\log\frac{2}{\delta}}{2T_n(a)}} \geq \mu(a^*)$$

- Reordering *[math]*

$$T_n(a) \leq \frac{2\log\frac{2}{\delta}}{\Delta(a)^2}$$

under event $\mathcal{E}$ and thus with probability $1 - nA\delta$

- Moving to the expectation *[statistics]*

$$\mathbb{E}[T_n(a)] = \mathbb{E}[T_n(a)|\mathcal{E}] + \mathbb{E}[T_n(a)|\mathcal{E}^C]$$

$$\mathbb{E}[T_n(a)] \leq \frac{2\log\frac{2}{\delta}}{\Delta(a)^2} + n(nA\delta)$$

- Trading-off the two terms $\delta = \frac{1}{n^2}$, we obtain:

$$\mathbb{E}[T_n(a)] \leq \frac{4\log 2n}{\Delta(a)^2} + A$$

Tuning the ρ Parameter

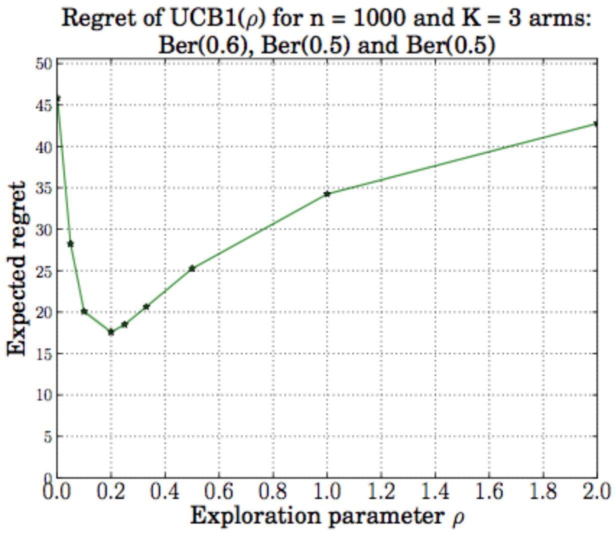
Recall:

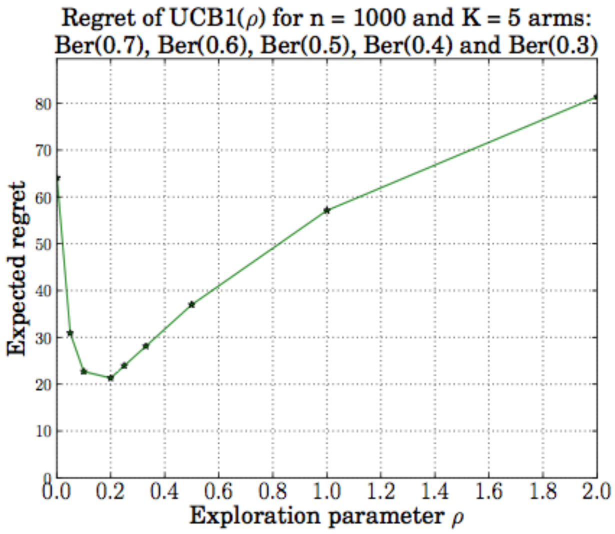
$$a_{t+1} = \arg\max_i \hat{\mu}_t(a) + \rho \sqrt{\frac{\log \frac{2}{\delta_t}}{2T_t(a)}}$$

Theory

- $\rho < 1$, polynomial regret w.r.t. n
- $\rho \geq 1$, logarithmic regret w.r.t. n

Practice: $\rho = 0.2$ is often the best choice



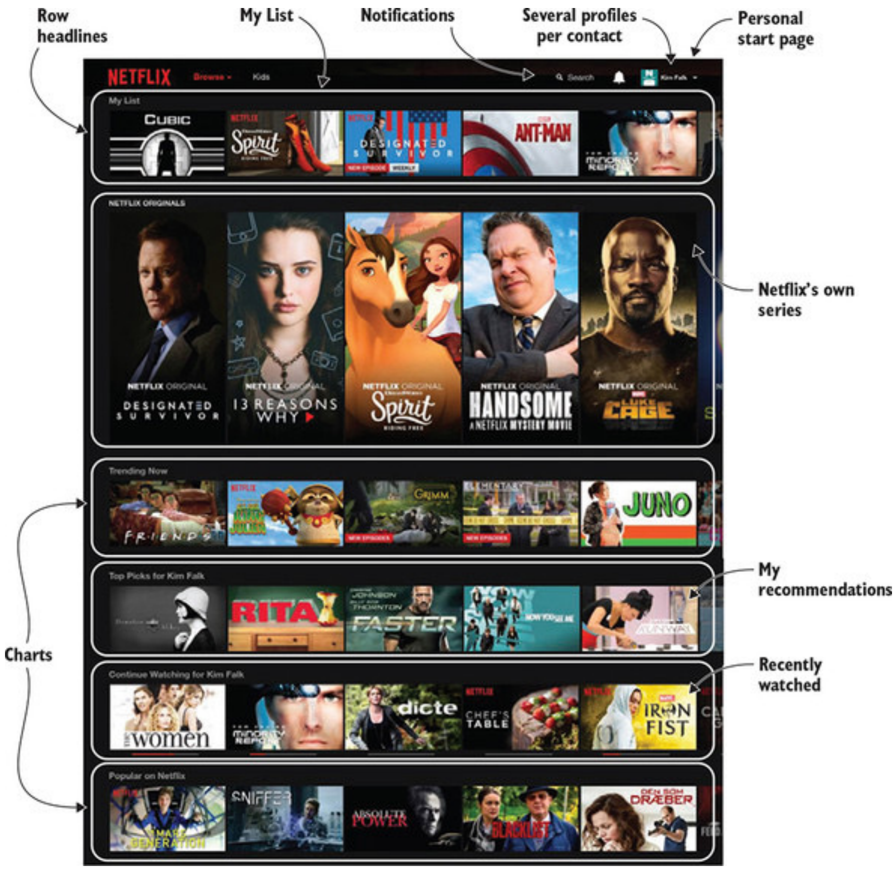


Outline

1. From RL to bandits
2. Exploration Strategies
3. **Linear and contextual linear bandits**

A Simple Recommendation System

- A RS can recommend specific movies [Netflix has 3600 movies (vs 14 genres)]
- Users arrive at random and no information about the user is available
- The RS picks a movie to the user
- The feedback is whether the user watched the movie or not
- Objective: Design a RS that maximizes the number of movies watched



RS as a Multi-armed Bandit

for $i = 1, \dots, n$ **do**

1. User arrives
2. Recommend movie a_t
3. Reward

$$r_t = \begin{cases} 1 & \text{user watches movie } a_t \\ 0 & \text{otherwise} \end{cases}$$

Endfor

Issue: Too many movies are available to collect enough feedback for each movie separately

RS as a Linear Bandit

The *model*

- $\mu(a) = \mathbb{E}[r(a)]$ is the probability a random user watches movie a
- Each movie a is characterized by some features $\phi(a) \in \mathbb{R}^d$ (e.g. genre, release date, past rating, income, etc)
- Assumption:
  - The expected value is a linear function $\mu(a) = \phi(a)^T \theta^*$ (with $\theta^* \in \mathbb{R}^d$ unknown)
  - The rewards are noisy observations $r_t(a) = \mu(a) + \eta_t$ with $\mathbb{E}[\eta_t] = 0$

The *objective*

- Maximize sum of reward $\mathbb{E}[\sum_{t=1}^{n} r_t]$

Recall: UCB

1. Computation of estimates

$$\hat{\mu}_t(a) = \frac{1}{T_t(a)} \sum_{s=1}^{t} r_s \mathbb{I}\{a_s = a\}$$

2. Evaluation of uncertainty

$$|\hat{\mu}_t(a) - \mu(a)| \leq \sqrt{\frac{\log \frac{1}{\delta}}{T_t(a)}}$$

3. Mechanism to combine estimates and uncertainty

$$B_t(a) = \hat{\mu}_t(a) + \rho \sqrt{\frac{\log \frac{1}{\delta_t}}{T_t(a)}}$$

4. Select the best action (according to its combined value)

$$a_t = \arg\max_a B_t(a)$$

Issue: $T_t(a)$ is likely to be 0 for most a. We need more sample efficient estimates.

The Regret

$$R_n = \max_a \mathbb{E}\left[\sum_{t=1}^{n} r_t(a)\right] - \mathbb{E}\left[\sum_{t=1}^{n} r_t(a_t)\right]$$
$$= \mathbb{E}\left[\sum_{t=1}^{n} \big(\phi(a^*) - \phi(a_t)\big)^T \theta^*\right]$$

Issue: a^* unlikely to be ever selected if $n \ll A$

Least-Squares Estimate of θ^*

- Least-squares estimate

$$\hat{\theta}_t = \arg\min_{\theta \in \mathbb{R}^d} \frac{1}{t} \sum_{s=1}^{t} (r_s - \phi(a_s)^T \theta)^2 + \lambda \|\theta\|^2$$

- Closed form solution

$$A_t = \sum_{s=1}^{t} \phi(a_s)\phi(a_s)^T + \lambda I \qquad b_t = \sum_{s=1}^{t} \phi(a_s) r_s$$

$$\Longrightarrow \hat{\theta}_t = A_t^{-1} b_t$$

- Estimate of value of action a

$$\hat{\mu}_t(a) = \phi(a)^T \hat{\theta}_t$$

- For analysis, need stronger concentration inequalities

Recipe of LinUCB

1. Computation of estimates

$$\hat{\theta}_t = A_t^{-1} b_t \qquad \hat{\mu}_t(a) = \phi(a)^T \hat{\theta}_t$$

2. Evaluation of uncertainty

$$|\hat{\mu}_t(a) - \mu(a)| \leq \alpha_t \sqrt{\phi(a)^T A_t^{-1} \phi(a)}$$

3. Mechanism to combine estimates and uncertainty

$$B_t(a) = \hat{\mu}_t(a) + \alpha_t \sqrt{\phi(a)^T A_t^{-1} \phi(a)}$$

4. Select the best action (according to its combined value)

$$a_t = \arg\max_a B_t(a)$$

LinUCB: Algorithm

for $t = 1, \ldots, n$ **do**

1. Compute upper-confidence bound
$$B_t(a) = \hat{\mu}_t(a) + \alpha_t \sqrt{\phi(a)^T A_t^{-1} \phi(a)}$$
2. Take action $a_t \arg\max_a B_t(a)$
3. Observe reward $r_t \sim \phi(a_t)^T \theta^* + \eta_t$
4. Update statistics for action a_t
$$A_{t+1} = A_t + \phi(a_t)\phi(a_t)^T$$
$$\hat{\theta}_{t+1} = \mathrm{A}_{\mathrm{t}+1}^{-1} b_{t+1}$$

endfor

LinUCB: Regret

Theorem

Consider a linear MAB problem with actions defined in $\mathbb{R}^d$ and unknown parameter $\theta^* \in \mathbb{R}^d$. If LinUCB is run with $\delta_t = \frac{1}{t}$ for n steps, then it suffers a regret:

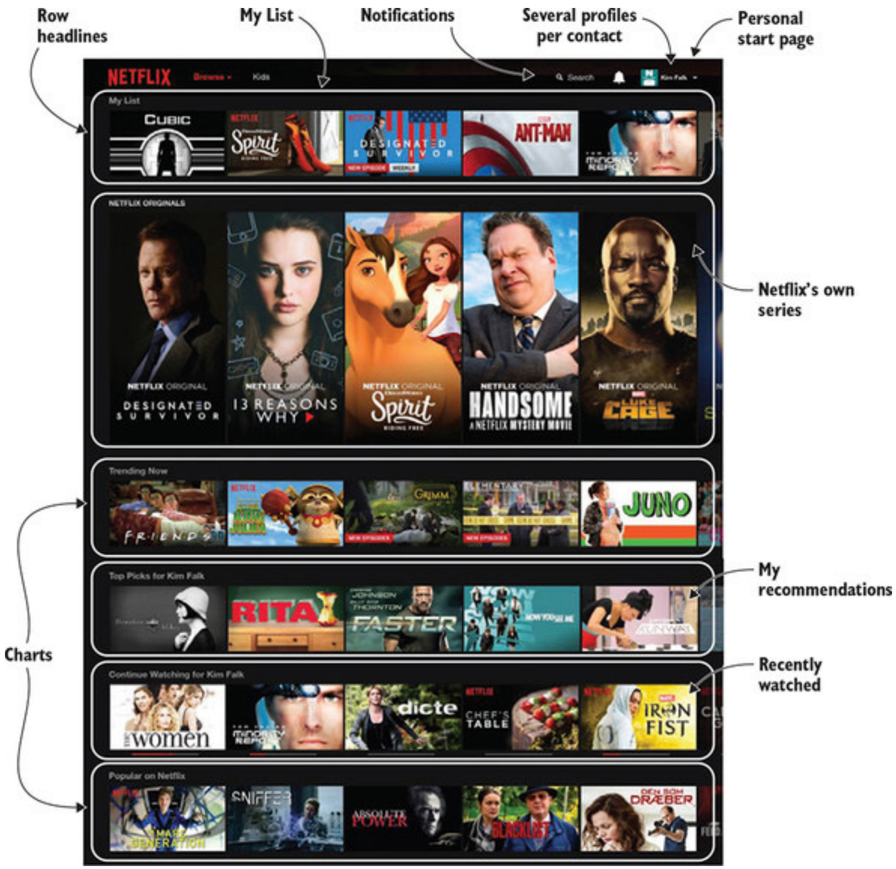
$$R_n = \mathcal{O}\left(d\sqrt{n \log n}\right)$$

- It depends on d but not the number of actions A
- If $A < \infty$, we can improve the bound to

$$R_n = \mathcal{O}\left(\sqrt{dn \log(nA)}\right)$$

A Simple Recommendation System

- A RS can recommend specific movies [Netflix has 3600 movies (vs 14 genres)]
- Users arrive at random and we have information about them
- The RS picks a movie to the user
- The feedback is whether the user watched the movie or not
- Objective: Design a RS that maximizes the number of movies watched



RS as a Multi-armed Bandit

for $i = 1, \ldots, n$ **do**

1. User arrives u_t
2. Recommend movie a_t
3. Reward

$$r_t = \begin{cases} 1 & \text{user watches movie } a_t \\ 0 & \text{otherwise} \end{cases}$$

Endfor

Issue: Too many users to collect enough feedback for each user separately

RS as a Contextual Linear Bandit

The *model*

- $\mu(u,a) = \mathbb{E}[r(u,a)]$ is the probability user u watches movie a
- Each user u and movie a is characterized by some features $\phi(u,a) \in \mathbb{R}^d$ (e.g. name, location, genre, release date, past rating, income, etc)
- Assumption:
  - The expected value is a linear function $\mu(u,a) = \phi(u,a)^T \theta^*$ (with $\theta^* \in \mathbb{R}^d$ unknown)
  - The rewards are noisy observations $r_t(u,a) = \mu(u,a) + \eta_t$ with $\mathbb{E}[\eta_t] = 0$

The *objective*

- Maximize sum of reward $\mathbb{E}[\sum_{t=1}^{n} r_t]$

Theory: $R_n = \mathcal{O}\left(d\sqrt{n \log n}\right)$

Summary & takeaways

- It is possible to determine the best action directly from data & interaction (i.e. model free), rather than through explicit modeling of the problem.
- The trade-off between exploration and exploitation can be formalized as regret; it a pervasive theme in reinforcement learning, and is already observed in multi-armed bandits.
- Multi-armed bandits are state-less decision problems. Contextual bandits have a state, but states are drawn i.i.d., rather than dependent on the past. Both can be solved to optimal regret (modulo log factors).
- MAB and CB have wide applications in recommendation systems, ad choice, health advice, education, etc.
- Optimism under uncertainty is an adaptive exploration strategy which optimally balances exploration and exploitation.

Further Reading

1. T. Lattimore and C. Szepesvári, *Bandit algorithms*. Cambridge University Press, 2020.

References

1. Alessandro Lazaric. INRIA Lille. Reinforcement Learning. 2017, Lecture 6.

2. Aleksandrs Slivkins. Introduction to Multi-Armed Bandits. 2019. Chapters 1, 8.