DATA 37200: Learning, Decisions, and Limits
(Winter 2026)

Lecture 6: Intro to contextual bandits

Instructor: Frederic Koehler

References

Lattimore and Szepasvári chapter 18. Foster-Rakhlin lecture notes, chapter 3.

Beyond the Multi-Armed Bandit

- **Standard MAB:** You have k slot machines. One is the best on average. You find it and stick to it.
- **The Missing Piece:** In the real world, we often have **side information** before we make a choice.
- **Example:**
  - In MAB, you recommend the same "best" movie to every user.
  - In **Contextual Bandits**, you look at the user's history (Context) before choosing which movie to show.
- In medicine, some people are allergic to penicillin, drugs may have interactions, ... so you obviously need to customize your suggestion to the person at hand.
- In sports gambling, you may want to bet differently depending on the game, players, horse, ...

The Interaction Protocol

For each round $t = 1, \ldots, T$:

1. **Observe Context:** The environment reveals $x(t) \in \mathcal{X}$.
2. **Choose Action:** The agent selects an arm $i(t) \in \{1, \ldots, k\}$.
3. **Receive Reward:** The agent receives $r(t) \in [0, 1]$ sampled from:
$$r(t) \sim D_{i(t)}(\cdot \mid x(t))$$
4. **Feedback:** We only see the reward for the chosen arm $i(t)$. We do *not* see what would have happened if we picked a different arm.

Notation

$D_1(\cdot \mid x(t)), \ldots, D_k(\cdot \mid x(t))$ are the k reward distributions conditioned on the current context.

Real-World Examples

Application	**Context** $x(t)$	**Arms** $i(t)$
Personalized Med.	Patient Vitals/Genetics	Different Drugs
News Feed	User Browsing History	Articles to Show
Ad Placement	Search Query/Location	Specific Ad Banner
Mobile Health	Time of day/Step count	Push Notification

Crucially: The "best" arm changes as $x(t)$ *changes.*

The Benchmark: The Optimal Policy

In standard MAB, the benchmark is a single best arm i^*. In Contextual Bandits, the benchmark is a **Policy** $\pi : \mathcal{X} \to [k]$.

- Let $f^*(x, i) = \mathbb{E}[r \mid x, i]$ be the expected reward.
- The best possible action for a specific context x is:

$$i^*(x) = \arg \max_{i \in \{1,\ldots,k\}} f^*(x, i)$$

- The **Optimal Policy** π^* is the mapping that always chooses $i^*(x)$ for any given x.

Defining Regret

Regret measures how much reward we "lost" by not being perfect.

Contextual Regret

$$\text{Regret}(T) = \sum_{t=1}^{T} \mathbb{E}[r(t) \mid x(t), i^*(x(t))] - \sum_{t=1}^{T} \mathbb{E}[r(t) \mid x(t), i(t)]$$

- **In MAB:** We compare ourselves to the best *fixed* arm.
- **In Contextual Bandits:** We compare ourselves to the best *mapping* from contexts to arms.
- This is a much "harder" benchmark!

The Challenge of Generalization

- If $x(t)$ is unique every time (e.g., a continuous vector), we might **never see the same context twice**.
- We cannot simply "average" the rewards for arm 1 like we do in MAB.
- We must **generalize**: If arm 1 was good for context x, is it also good for context x'?
- This requires assuming/modeling a relationship between contexts and rewards (e.g., a function class $\mathcal{F}$ such as linear or neural networks, which predicts the rewards of the arms).

A baseline approach

If the number of possible contexts is limited, we *can* solve the problem fairly directly using MAB.

- Let $\mathcal{C}$ be the context space and $|\mathcal{C}|$ the number of contexts.
- For every context $x \in \mathcal{C}$, use one MAB as an "expert" on this context.
- If T_x is the number of times context x appears, then

$$\mathit{Regret} \lesssim \sum_{x \in \mathcal{C}} \sqrt{K T_x \log(T_x)} \lesssim \sqrt{K T \log(T)}$$

- Here we used Cauchy-Schwarz: $\sum_i a_i b_i \leq \sqrt{\sum_i a_i^2} \sqrt{\sum_i b_i^2}$.

Modeling beyond the baseline

How to model the problem when $\mathcal{C}$ is too big to enumerate?

- Let

$$f^*(x, i) = \mathbb{E}[r(t) \mid x(t) = x, i(t) = i] = \mathbb{E}_{r \sim D_i(\cdot|x)}[r]$$

  be the *expected reward of arm i* given context x.
- f^* is unknown, *however* we assume knowledge of a class of functions

$$\mathcal{F} : \mathcal{X} \times [K] \to [0, 1]$$

  such that $f^* \in \mathcal{F}$.
- Contextual bandit combines *learning*/forecasting (trying to figure out the true reward function f^*) with *decision-making* (how to pick the arm to pull).

Can we just Explore-Then-Commit (ETC)?

- ETC was an easy, but suboptimal, method for solving the MAB. Can we use it for CB?
- Natural ETC approach: use the first m rounds to learn the reward model f^* by picking random actions, fit a good model $\hat{f}$ based on observations, and then be greedy according to $\hat{f}$ forecasts.
- Given first m observations, we can try to use any supervised learning method to learn $\hat{f}$ from data.
- I.e. after m rounds, pick arm

$$i(t) = \arg\max_i \hat{f}(x(t), i).$$

- Does it work?

Failure of ETC in this model

- As formulated, ETC/NAIVE-EE (Explore Then Commit) is not a good strategy for our model.
- This is because the contexts $x(1), \ldots, x(T)$ are arbitrary. So if you explore for the first m rounds, I can show you only one context in the first m rounds.
- ETC would be okay if we modeled contexts $x(1), \ldots, x(T)$ as i.i.d. samples.
- More general formulation is nice in that contexts can change over time — realistic concern. (Ex: no users with iphones before 2007, lots of users with iphones by now).
- However, we assume the true reward model f^* *does not* change over time.

Future lectures on CB

- Following the Foster-Rakhlin notes, we will cover *two* quite different approaches to solving the CB.
- Approach 1: generalize UCB approach (Upper Confidence Bounds). "optimism"
- Approach 2: modularize forecasting and decision making. Ex: ϵ-greedy and smarter variants.
- We will start with approach 2 first. (simpler?)
- CB is a nice special case of RL where we do not model *the effect of our interactions on the environment.* Practically and theoretically clean.