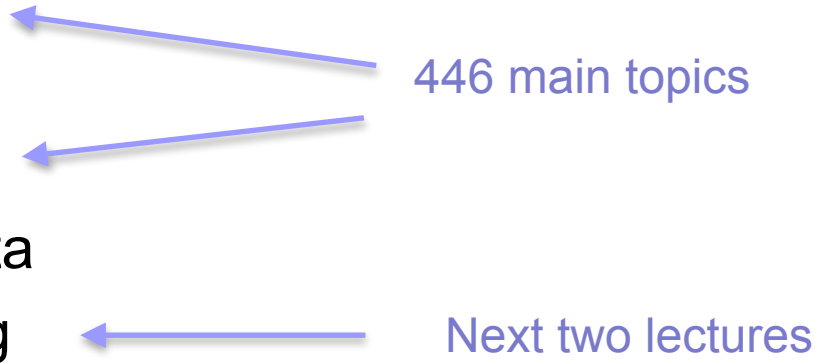
CSE 446 Bandits

Leo Maynard-Zhang

UNIVERSITY of WASHINGTON

Categories of ML

- Supervised Learning
  - Offline labeled data
- Unsupervised Learning
  - Offline unlabeled data
- Reinforcement Learning
  - Online sequential decision-making



What next?

- After 446:
  - CSE 493S/599S: Advanced ML - Jamie Morgenstern/ Sewoong Oh
    - Next Autumn/Spring
  - CSE 541: Interactive Learning - Kevin Jamieson
    - Next Spring
  - CSE 542: Reinforcement Learning - Kevin Jamieson
    - Next Winter
  - CSE 543: Deep Learning - Simon Du
    - Next Autumn
  - CSE 599J: Social Reinforcement Learning - Natasha Jaques
    - Next Autumn

Multi-armed bandits

Decision-making in the face of uncertainty

- In life we are faced with many difficult decisions
- Often these decisions have uncertain outcomes
  - “How much time should I spend studying for my exam?”
  - “Which stock should I invest in?”
  - “Should I pay the parking fee?”
- We have had to make such decisions likely more than once in our lifetime
- How do we find a way to make the optimal choice when we aren’t certain about the outcome?
- ML approach: What if we used the outcomes of our past decisions to help inform our future ones?

Slot machines

- Your friend who owns a casino says you can come play his 10 slot machines for free, but you only get 100 pulls
- Each slot machine has a different unknown expected payoff
  - Payoffs follow some distribution, for example, Gaussian
- Assume one of these slot machines has a higher expected payoff than all the others
- How will you maximize your winnings?

Slot machines

- One approach:
  - Play each of the 10 slot machines once.
  - Then commit to the slot machine with the highest return, and play it 90 times.
- What is the issue with this approach?
  - Very volatile!
  - The slot machines are random, and the one that initially gives the highest return might not actually be the best one
  - Worst case: Play suboptimal slot machines 91 + 8 = 99 times!!!

Slot machines

- Second approach:
  - Play each of the 10 slot machines 9 times
  - Then commit to the slot machine with the highest average return, and play it 10 times.
- What is the issue with this approach?
  - Too safe!
  - Best case: Play best slot machine only 10 + 9 = 19 times!!!

Exploration vs Exploitation

- We must balance the following tradeoff:
  - Exploration: We want to find the slot machine that is optimal
    - This means we have to explore every slot machine, including those that are suboptimal
  - Exploitation: We want to play the slot machine that we believe to be optimal
    - This means we have to concentrate on the slot machines that have already given favorable returns
- These objectives are clearly conflicting

The bandit problem

- There are k possible actions enumerated as $\{1,2,...,k\}$, with a distribution for each action dictating rewards: $(\mu_1, \ldots, \mu_k)$
  - k slot machines
  - μ_i denotes the expected return of the ith slot machine
- We are allowed to do n actions sequentially
  - n total pulls
- How do we make the right decisions in such uncertain environments?

First algorithm: Explore-then-commit

- First choose hyperparameter m
- Then run the following algorithm:
  - Input $k, n, m \in \mathbb{N}$ and unknown $(\mu_1, \ldots, \mu_k)$
  - For $i = 1,...,k$:
    - Play action i m times, observe m rewards $r_1, \ldots, r_m$ sampled from μ_i and compute empirical mean $\hat{\mu}_i = \frac{1}{m} \sum_{j=1}^{m} r_j$
  - For $t = mk, \ldots, n$:
    - Play action $a_* = \arg\max_{i \in [k]} \hat{\mu}_i$

This is just the slot machine algorithm from earlier defined formally.

It turns out we can do much better than this, see UCB and Thompson Sampling.

Today's lecture

- We will be focusing on the setting where our action space is featurized in some finite dimensional space
- That is, instead of our action set being $\{1,2,...,k\}$, it will be of the form $\mathcal{X} \subset \mathbb{R}^d$
  - $\mathcal{X}$ can be infinite!
- The reward distributions will be dictated by some common linear model
- Why?
  - Think about driving a car. Is there a way to describe the set of available actions in a finite and one-dimensional way?

Linear bandits

Recap: Linear regression

- Given fixed dataset $\{x_i, y_i\}_{i=1}^n$, with $y_i = x_i^\top w + \epsilon_i$, where $w \in \mathbb{R}^d$ is unknown and ϵ_i is zero-mean Gaussian noise.
- Goal: estimate w to make predictions on unseen data
- Ridge estimation: $\widehat{w} = \arg\min_w \sum_{i=1}^n (x_i^\top w - y_i)^2 + \lambda ||w||_2^2$
- Closed form: $\widehat{w} = \left(\sum_{i=1}^n x_i x_i^\top + \lambda I \right)^{-1} \sum_{i=1}^n x_i y_i$

Exploring Mars

- We want to analyze different rock samples on mars, and figure out which ones have the highest concentration of (water) ice
- The landscape is full of different types of rocks, with different size, colors, hardness, etc.
  - We assume there is a linear relationship between these attributes and the amount of ice contained in the rock
- Every hour we choose a rock who's ice content we want to analyze, which is a time consuming process
- After analyzing many rocks, our trip comes to an end. We need to report back to earth: what kind of rocks have the highest amount of ice?

Linear bandit setting (Pure exploration)

- Input action set $\mathcal{X} \subset \mathbb{R}^d$, $n \in \mathbb{N}$, and unknown $w_* \in \mathbb{R}^d$
- For $t = 1,...,n$: ← Data chosen sequentially
  - Choose $x_t \in \mathcal{X}$ based on $\{x_i, y_i\}_{i=1}^{t-1}$
  - Observe $y_t = x_t^\top w_* + \epsilon_t$ ← ϵ_t is unobserved zero-mean noise
- Return $\hat{x} \in \mathcal{X}$ based on $\{x_i, y_i\}_{i=1}^{n}$

Goal: Return $x_* = \arg\max_{x \in \mathcal{X}} x^\top w_*$ with high probability

Reward Maximization

- What if our goal is to maximize reward? i.e. maximize $\sum_{t=1}^{n} y_t$
- This is the standard objective in sequential decision-making
- We now have to be more careful about how we choose actions at each time step—every action we make now contributes directly to our objective
  - Have to now consider the "cost" of our actions
  - Cannot just explore freely
- Example: Advertising
  - Each ad we place generates some revenue
  - Want to maximize our total revenue

Exploration vs Exploitation

- We must balance the following tradeoff:
  - Exploration: We want to find the action that is optimal
    - This means we have to explore every action, including those that are suboptimal
  - Exploitation: We want to play the action that we believe to be optimal
    - This means we have to concentrate on actions that have already given favorable returns
- These objectives are clearly conflicting
- How to measure how well an algorithm balances this tradeoff?

Regret

- Let π be a bandit algorithm, i.e. a decision rule that at time step t maps history $\{x_i, y_i\}_{i=1}^{t-1}$ to action $x_t \in \mathcal{X}$
  - We allow π to be deterministic or stochastic
- We define the (pseudo) regret of algorithm π on problem instance w_* with interaction length n as

$$R_n(\pi, w_*) = \sum_{t=1}^{n} \max_{x \in \mathcal{X}} x^\top w_* - x_t^\top w_*$$

We will use R_n for shorthand

Max possible reward

Actual reward obtained by π

Regret

$$R_n = \sum_{t=1}^{n} \max_{x \in \mathcal{X}} x^\top w_* - x_t^\top w_*$$

- What is a good value of regret?
- We want our regret to satisfy $R_n = o(n)$
  - $R_n = o(n) \iff \lim_{n \to \infty} \frac{R_n}{n} = 0$
- Average regret goes to 0 in the limit
  - Our algorithm eventually only plays optimal actions
- Examples: $O(\sqrt{n}), O(\log n), O(\sqrt{n} \log n)$
  - "sublinear"
  - $n \neq o(n)$

Linear bandit setting (Regret minimization)

- Input action set $\mathcal{X} \subset \mathbb{R}^d$, $n \in \mathbb{N}$, and unknown $w_* \in \mathbb{R}^d$
- For $t = 1,...,n$:
  - Choose $x_t \in \mathcal{X}$ based on $\{x_i, y_i\}_{i=1}^{t-1}$
  - Observe reward $y_t = x_t^\top w_* + \epsilon_t$

Goal: Minimize regret $R_n = \sum_{t=1}^{n} \max_{x \in \mathcal{X}} x^\top w_* - x_t^\top w_*$

This is the same as maximizing reward!

Example: Spotify

- Suppose we are choosing songs to autoplay for a Spotify user. After the user is done listening to a song we choose a song to play next. Then we measure how long the user listened to the song and whether or not they saved it.
  - n: number of songs the user plans to listen to
  - d: BPM, genre, popularity, release date, etc.
  - $\mathcal{X}$: different songs we can recommend
  - y_t: percentage of song listened to + whether or not the user saved it.
- Goal: recommend songs that the user enjoys listening to, so they continue to use our app

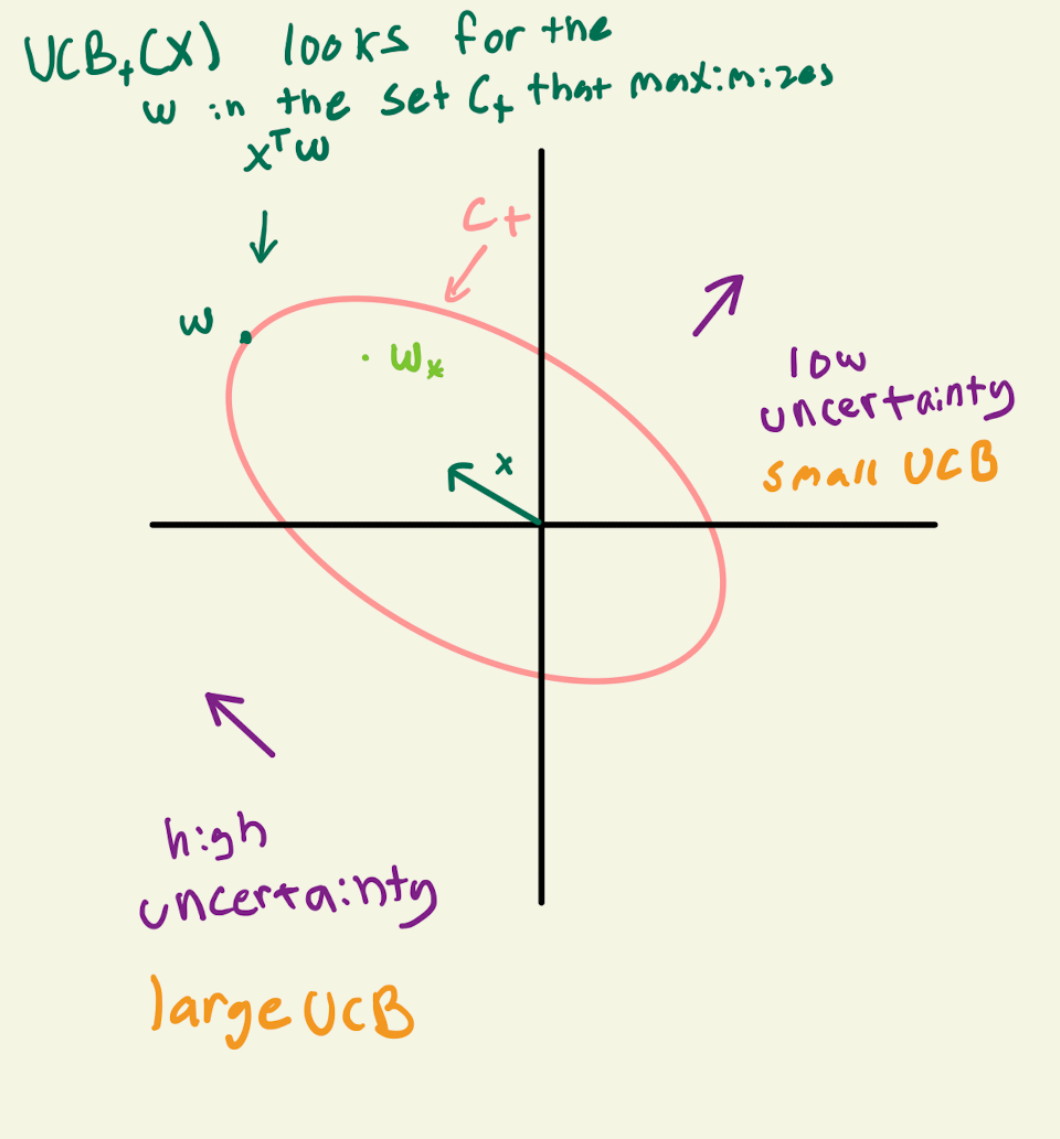
Optimism

- How do we tackle the exploration-exploitation problem?
- “Optimism in the face of uncertainty”
- Suppose you visit the same restaurant each week for Sunday brunch
  - If you only order the same dishes every week, you may never discover some dishes that you might love!
  - Solution: have an “optimistic” view of each menu item
    - Assume you’ll really like menu items you haven’t tried
    - This will lead to you trying every dish until you have sufficient reason to believe you don’t like them
- Key idea: Overestimating the true value of each action leads to efficient exploring

Optimism: Upper Confidence Bound

- Idea: at each time step t construct confidence set C_t that contains w_* with high probability
- For each $x \in \mathcal{X}$, define $\text{UCB}_t(x) = \max_{w \in C_t} x^\top w$
- Notice if $w_* \in C_t$ then $\text{UCB}_t(x) \geq x^\top w_*$ for all $x \in \mathcal{X}$
  - $\text{UCB}_t(x)$ is an overestimate of the true value of each $x \in \mathcal{X}$!

Optimism: Upper Confidence Bound



Optimism: Upper Confidence Bound

- There is an equivalent, possibly more familiar form of UCB, which sets $\text{UCB}_t(x) = x^\top \widehat{w} + b_t$, where $\widehat{w}$ is the ridge estimator, and b_t is some carefully chosen "exploration bonus"
- How to choose b_t?
  - Notice:

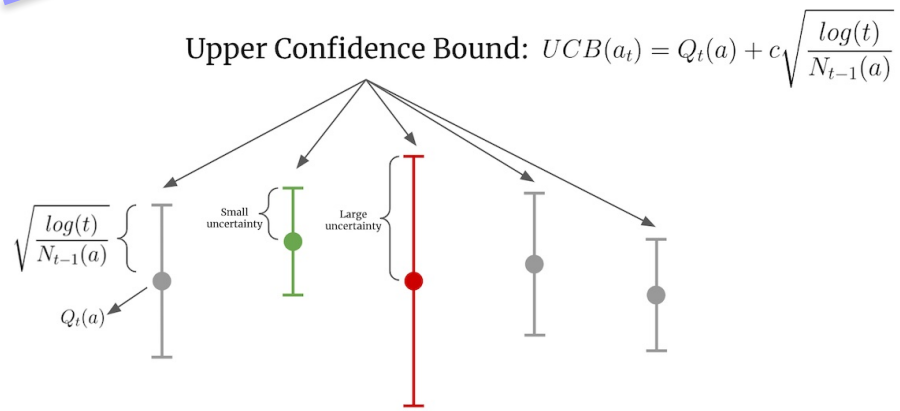
$$x^\top w_* = x^\top \widehat{w} + x^\top (w_* - \widehat{w})$$
$$\leq x^\top \widehat{w} + ||x||_{V^{-1}} ||w_* - \widehat{w}||_V$$
$$\leq x^\top \widehat{w} + \gamma_t \cdot ||x||_{V^{-1}}$$

$V = \sum_{i=1}^{t-1} x_i x_i^\top$

b_t

γ_t is some high probability upper bound, since w_* is unknown

(visual corresponds to the k-armed case, but the intuition is the same)



LinUCB Algorithm

- Input action set $\mathcal{X} \subset \mathbb{R}^d$, $n \in \mathbb{N}$, and unknown $w_* \in \mathbb{R}^d$
- For $t = 1,...,n$:
  - Construct C_t based on $\{x_i, y_i\}_{i=1}^{t-1}$
  - Denote $\text{UCB}_t(x) = \max_{w \in C_t} x^\top w$
  - Choose $x_t = \arg\max_{x \in \mathcal{X}} \text{UCB}_t(x)$
  - Observe $y_t = x_t^\top w_* + \epsilon_t$

Confidence set

- Fix $t \in [n]$
- We will have some fixed data set $\{x_i, y_i\}_{i=1}^{t-1}$
  - Suppose for now that this data set is collected non-adaptively
- Let $V = \sum_{i=1}^{t-1} x_i x_i^\top$
- Let $\widehat{w}$ be the LS estimator trained on this data
- We then have that $\mathbb{E}[\,||w_* - \widehat{w}||_V^2\,] \leq d$
  - This suggests that for our confidence set to contain w_*, we should construct an ellipsoid around the LS estimator with "radius" of order $\sqrt{d}$

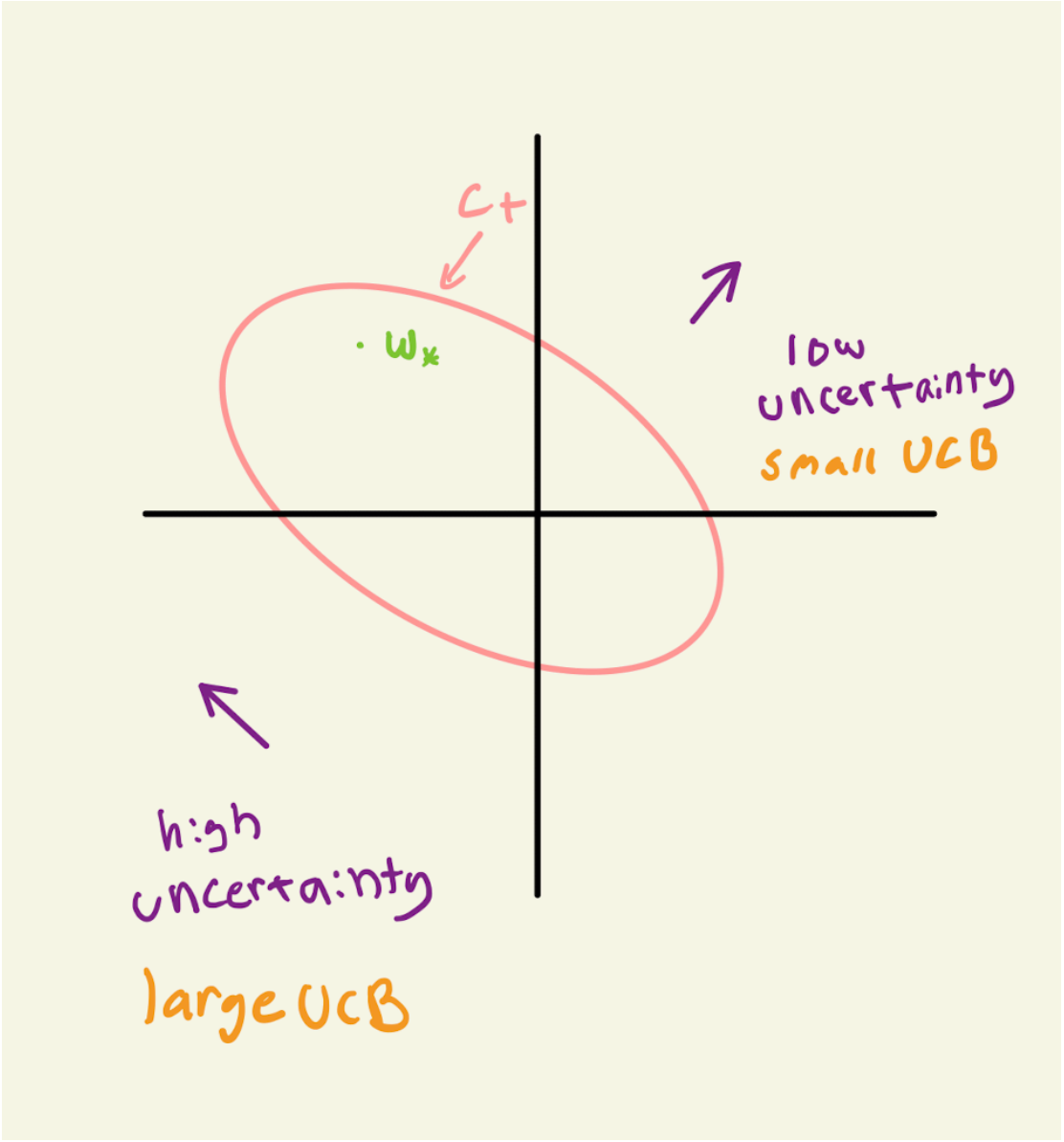
Confidence set

- Denote $V_t = \sum_{i=1}^{t} x_i x_i^\top + \lambda I$ for some $\lambda > 0$

Ridge estimator!

- Let $\widehat{w}_t = V_t^{-1} \sum_{i=1}^{t} x_i y_i$
- Confidence interval construction:
  - Choose confidence $\delta > 0$
  - Set $\gamma_t = O\left(\sqrt{\log\left(\frac{1}{\delta}\right) + d \log n}\right)$
  - Construct $C_t = \left\{ w \in \mathbb{R}^d : || \widehat{w}_{t-1} - w ||_{V_{t-1}} \leq \gamma_t \right\}$
  - Then with probability $\geq 1 - \delta$, $w_* \in C_t$ for all $t \in [n]$ (Theorem 2 Abbasi-Yadkori 2011)

Confidence set



LinUCB Regret Analysis

- (Theorem 19.2 Lattimore & Szepesvári 2020) with probability at least $1 - 1/n$, the worst-case regret of LinUCB satisfies:

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

LinUCB Regret Analysis

The following are tools needed to prove the theorem:

- Elliptical potential lemma: $\sum_{t=1}^{n} ||x_t||^2_{V_{t-1}^{-1}} \leq O\left(d \log n\right)$
  - (Lemma 11 Abbasi-Yadkori 2011)

- Cauchy-Schwarz inequality: $a^\top b \leq ||a|| \cdot ||b||$
  - Corollaries:
    - $a^\top b \leq ||a||_A \cdot ||b||_{A^{-1}}$ for positive-definite A
    - $\sum_{i=1}^{n} a_i \leq \sqrt{n \sum_{i=1}^{n} a_i^2}$

LinUCB Regret Analysis

$$R_n = \sum_{t=1}^{n} \max_{x \in \mathcal{X}} x^\top w_* - x_t^\top w_*$$

Proof of Theorem 19.2:

1. Let $w_* \in \mathbb{R}^d$
2. Suppose $w_* \in C_t, \forall t \in [n]$ which happens with prob $\geq 1 - \delta$
3. Denote $x_* = \arg\max_{x \in \mathcal{X}} x^\top w_*$, and $\tilde{w}_t = \arg\max_{w \in C_t} x_t^\top w$
4. Since $x_*^\top w_* \leq \text{UCB}_t(x_*) \leq \text{UCB}_t(x_t) = x_t^\top \tilde{w}_t$, we have:

$$R_n = \sum_{t=1}^{n} x_*^\top w_* - x_t^\top w_* \leq \sum_{t=1}^{n} x_t^\top \tilde{w}_t - x_t^\top w_*$$

5. Apply Cauchy-Schwarz:

$$\sum_{t=1}^{n} x_t^\top \tilde{w}_t - x_t^\top w_* \leq \sum_{t=1}^{n} ||x||_{V_{t-1}^{-1}} ||\tilde{w}_t - w_*||_{V_{t-1}}$$

6. Use $||\tilde{w}_t - w_*||_{V_{t-1}} \leq ||\tilde{w}_t - \widehat{w}_{t-1}||_{V_{t-1}} + ||\widehat{w}_{t-1} - w_*||_{V_{t-1}} \leq 2\gamma_t$ to obtain:

$$\sum_{t=1}^{n} ||x||_{V_{t-1}^{-1}} ||\tilde{w}_t - w_*||_{V_{t-1}} \leq \sum_{t=1}^{n} ||x||_{V_{t-1}^{-1}} \cdot 2\gamma_t$$

LinUCB Regret Analysis

$$R_n = \sum_{t=1}^{n} \max_{x \in \mathcal{X}} x^\top w_* - x_t^\top w_*$$

Proof of Theorem 19.2 (continued):

7. Apply Cauchy-Schwarz:

$$\sum_{t=1}^{n} ||x||_{V_{t-1}^{-1}} \cdot 2\gamma_t \leq \sqrt{n \sum_{t=1}^{n} ||x||^2_{V_{t-1}^{-1}} \cdot 4\gamma_t^2}$$

8. Set $\delta = \frac{1}{n}$, plug in $\gamma_t = O(\sqrt{d \log n})$:

$$\sqrt{n \sum_{t=1}^{n} ||x||^2_{V_{t-1}^{-1}} \cdot 2\gamma_t^2} = \sqrt{n \cdot O(d \log n) \cdot \sum_{t=1}^{n} ||x||^2_{V_{t-1}^{-1}}}$$

9. Apply elliptical potential lemma:

$$\sqrt{n \cdot O(d \log n) \cdot \sum_{t=1}^{n} ||x||^2_{V_{t-1}^{-1}}} \leq \sqrt{n \cdot O(d \log n) \cdot O(d \log n)} = O(d\sqrt{n} \log n)$$

Near-optimality of LinUCB

- (Theorem 24.2 Lattimore & Szepesvári 2020) Suppose our action set $\mathcal{X}$ is the unit ball. Then for any algorithm π, there exists a problem instance $w_* \in \mathbb{R}^d$ such that:

$$R_n(\pi, w_*) \geq \Omega\left(d\sqrt{n}\right)$$

LinUCB is (in general) optimal up to logarithmic terms!

Contextual bandits

Contextual bandits

- The assumption that there is a singular “one-size fits all” best action is a bit naive
- At each time step we might have access to some contextual information that we can incorporate to make better decisions
  - These contexts can be thought of as a “state”
  - Some actions might be better than others depending on which “state” you are in

Recommender systems

- Contextual bandits are widely used by recommendation systems: Amazon, TikTok, etc.
- Context is the information available on each consumer
  - For example, demographic information: age, gender, geographic region, etc.
- We want to recommend things that the user will want to consume, but each user might have different preferences correlated with their demographic
  - For example teenage boys might want to watch video game clips, while older folks might want to watch cat videos.
  - Recommending the same content to both groups of people would not be wise!

Contextual bandit setting: linear approach

- Input action set $\mathcal{X}$, context set C, feature mapping $\phi : C \times \mathcal{X} \to \mathbb{R}^d$, $n \in \mathbb{N}$, and unknown $w_* \in \mathbb{R}^d$
- For $t = 1,...,n$:
  - Nature reveals context $c_t \in C$
  - Choose $x_t \in \mathcal{X}$ based on $\{x_i, y_i, c_i\}_{i=1}^{t-1}$ and c_t
  - Observe reward $y_t = \phi(c_t, x_t)^\top w_* + \epsilon_t$

Goal: Minimize regret $R_n = \sum_{t=1}^{n} \max_{x \in \mathcal{X}} \phi(c_t, x)^\top w_* - \phi(c_t, x_t)^\top w_*$

Reduces to a linear bandit. LinUCB is (an) option

Best action is changing at each time step

Summary

- Online decision-making
  - Data is no longer fixed
  - Action-feedback loop
- Reward maximization
  - Regret minimization
- Exploration vs. exploitation
  - Why is reward maximization hard?
- Optimism
  - UCB
- Contextual bandits
  - Incorporate contextual information to make more informed decisions

Further learning

- CSE 541: Interactive Learning; with Professor Kevin Jamieson
  - Multi-armed bandits
  - Linear bandits
    - Experimental design
    - Pure exploration
  - Contextual bandits
- Bandit Algorithms by Lattimore & Szepesvári
  - Textbook used in 541