TTIC 31290 - Machine Learning for Algorithm Design (Fall 2025)

Avrim Blum and Dravyansh Sharma

Lecturer: Avrim Blum

Lecture 14:
ML for Mechanism Design
and
The Adversarial Multi-armed Bandit Problem

ML for Mechanism Design

- In Lecture 11, we introduced the notion of incentive-compatible mechanisms.
- And we analyzed the sample complexity for learning *reserve prices* to maximize revenue in a Vickrey auction.
- Today, we will examine some other selling mechanisms.
- Material from: Balcan MF, Sandholm T, Vitercik E. “A general theory of sample complexity for multi-item profit maximization.” EC-2018, Operations Research 2025.

Posted-Price Mechanisms

- Selling n_{items} different items.
- Mechanism assigns each item i a price p_i.
- Buyers have arbitrary valuation functions over bundles $B \subseteq \{1, \ldots, n_{items}\}$, and purchase the bundle of highest value(B) – price(B).
- Assume some distribution D over buyers. How many samples do we need so that pricing that maximizes revenue on the sample is near-optimal over D?
- Analyze the pseudo-dimension.

Pseudo-dimension of posted price mechanisms

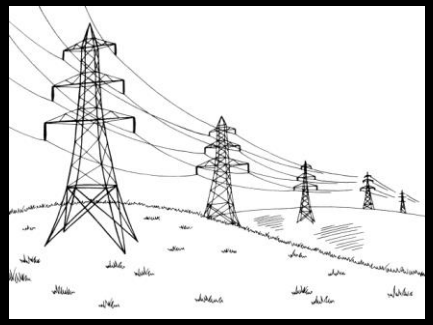
- Selling n_{items} different items.
- Mechanism assigns each item i a price p_i.

Dual space has dimension $d = n_{items}$.

- For any given buyer, preference of bundle B_1 vs B_2 determined by value(B_1) – price(B_1) vs value(B_2) – price(B_2).
- So, at most $k = \binom{2^{n_{items}}}{2}$ linear boundary functions.
- And within each region, revenue is linear.
- Gives a pseudo-dim of $O(n_{items} \cdot n_{items}) = O(n_{items}^2)$.

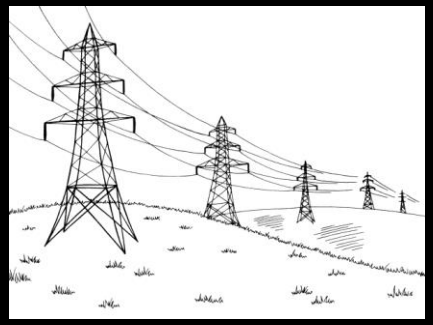
Two-part tariffs

- Single item, but multiple copies. (Assume any given buyer wants at most K).
- Seller sets an upfront fee p_0 and a per-unit fee p_1. Cost to buy k units is $p_0 + kp_1$.
- Examples: Costco, utilities, some amusement parks.



Menus of two-part tariffs

- Have L different two-part tariffs $(p_0^1, p_1^1), (p_0^2, p_1^2), \ldots (p_0^L, p_1^L)$
- Buyer picks a tariff i and a number $k \leq K$ of units to buy, and pays $p_0^i + kp_1^i$.
- What can we say about the pseudo-dimension now?



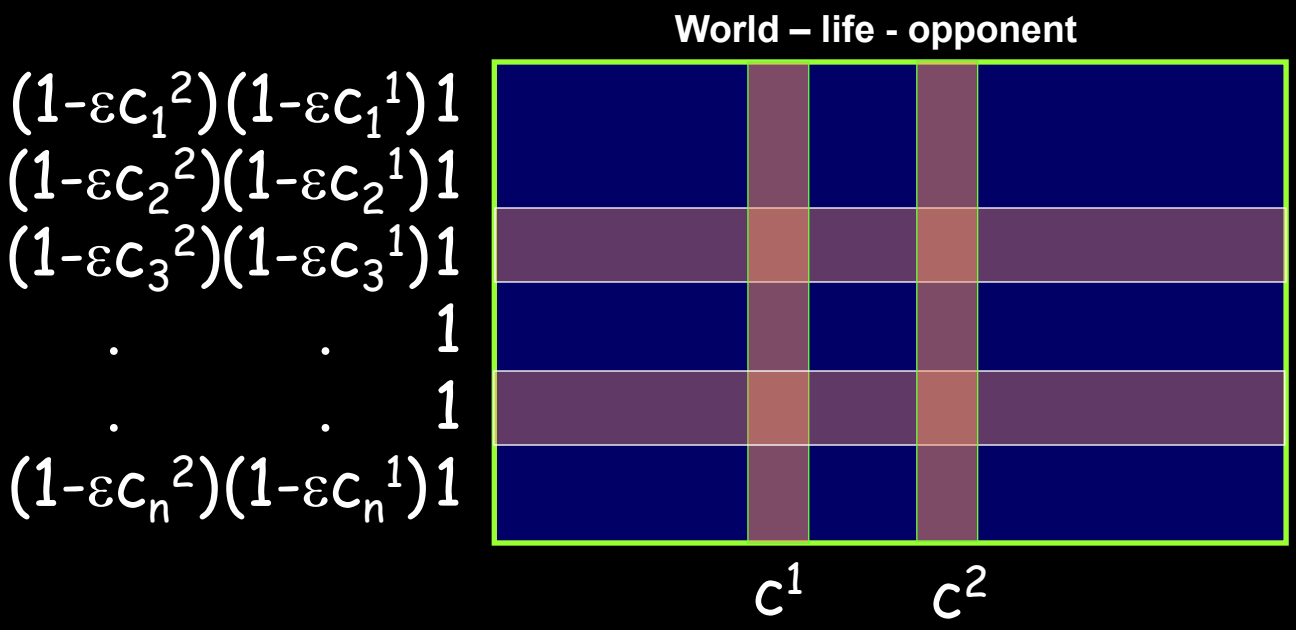
Pseudo-dimension of menus of two-part tariffs

- Have L different two-part tariffs $(p_0^1, p_1^1), (p_0^2, p_1^2), \ldots (p_0^L, p_1^L)$
- Buyer picks a tariff i and a number $k \leq K$ of units to buy, and pays $p_0^i + kp_1^i$.
- Dual space has dimension $d = 2L$.
- What does the boundary look like between choices (i_1, k_1) and (i_2, k_2)? Linear: $p_0^{i_1} + k_1 p_1^{i_1} = p_0^{i_2} + k_2 p_1^{i_2}$
- At most $O(L^2K^2)$ linear boundary functions, and revenue is linear within each region.
- So, pseudo-dimension is $O(L \log(KL))$.
- See [Balcan-Sandholm-Vitercik] for more.

Now, switching to online learning

The Adversarial Multi-Armed Bandit Problem

- In our previous discussion of online learning, we assumed "full feedback": we see how well we would have done had we made a different choice.



The Adversarial Multi-Armed Bandit Problem

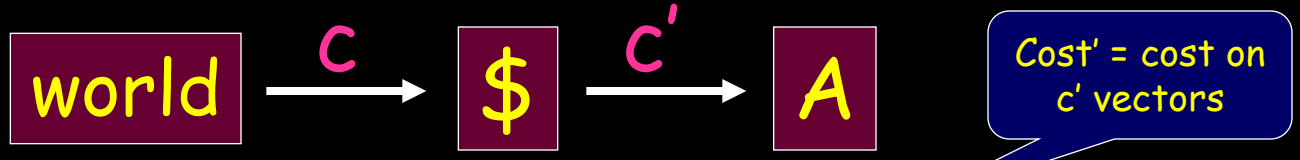
- In our previous discussion of online learning, we assumed "full feedback": we see how well we would have done had we made a different choice.
- But what if we only get feedback for the action we choose?
- This is called the "multi-armed bandit" setting.
- But first, a quick discussion of [0,1] vs {0,1} costs for RWM algorithm

[0,1] costs vs {0,1} costs.

We analyzed Randomized Wtd Majority for case that all costs in {0,1} (and slightly hand-waved extension to [0,1])

Here is an alternative simple way to extend to [0,1].

- Given cost vector c, view c_i as bias of coin. Flip to create boolean vector c′, s.t. $E[c'_i] = c_i$. Feed c′ to alg A.



- For any sequence of vectors c′, we have:
  - $E_A[\text{cost}'(A)] \leq \min_i \text{cost}'(i) + [\text{regret term}]$
- So, $E_\$[E_A[\text{cost}'(A)]] \leq E_\$[\min_i \text{cost}'(i)] + [\text{regret term}]$
- LHS is $E_A[\text{cost}(A)]$. (since A picks weights before seeing costs)
- RHS $\leq \min_i E_\$[\text{cost}'(i)] + [\text{r.t.}] = \min_i[\text{cost}(i)] + [\text{r.t.}]$

In other words, costs between 0 and 1 just make the problem easier…

Experts $\rightarrow$ Bandit setting

- In the bandit setting, only get feedback for the action we choose. Still want to compete with best action in hindsight.
- [ACFS02] give algorithm with cumulative regret $O((TN \log N)^{1/2})$. [average regret $O(((N \log N)/T)^{1/2})$.]

- Will do a somewhat weaker version of their analysis (same algorithm but not as tight a bound).

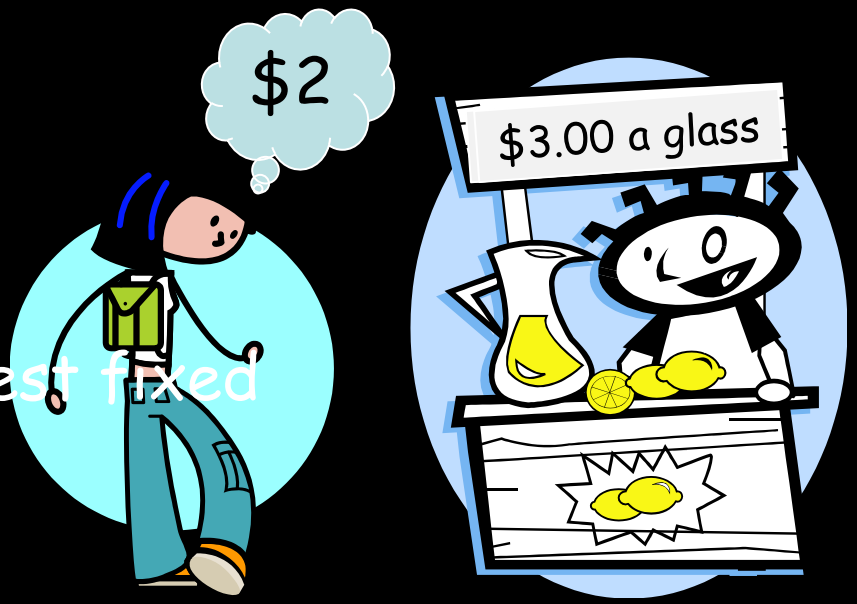
- Talk about it in the context of online pricing…

Online pricing

- Say you are selling lemonade (or bottles of water outside a football stadium).
- For t=1,2,…T
  - Seller sets price p^t
  - Buyer arrives with valuation v^t
  - If $v^t \geq p^t$, buyer purchases and pays p^t, else doesn't.
  - Repeat.

View each possible price as a different row/expert

- Assume all valuations $\leq h$.
- Goal: do nearly as well as best fixed price in hindsight.



- If v^t revealed, run RWM. E[gain] $\geq$ OPT$(1-\epsilon)$ - $O(\epsilon^{-1} h \log n)$.

Multi-armed bandit problem

Exponential Weights for Exploration and Exploitation (exp^3)

[Auer,Cesa-Bianchi,Freund,Schapire]

Expert $i \sim q^t$

Gain g_i^t

q^t
Exp3

Distrib p^t

Gain vector $\hat{g}^t$

RWM

n = #experts

$\widehat{OPT}$

$\leq nh/\gamma$

$q^t = (1-\gamma)p^t + \gamma$ unif

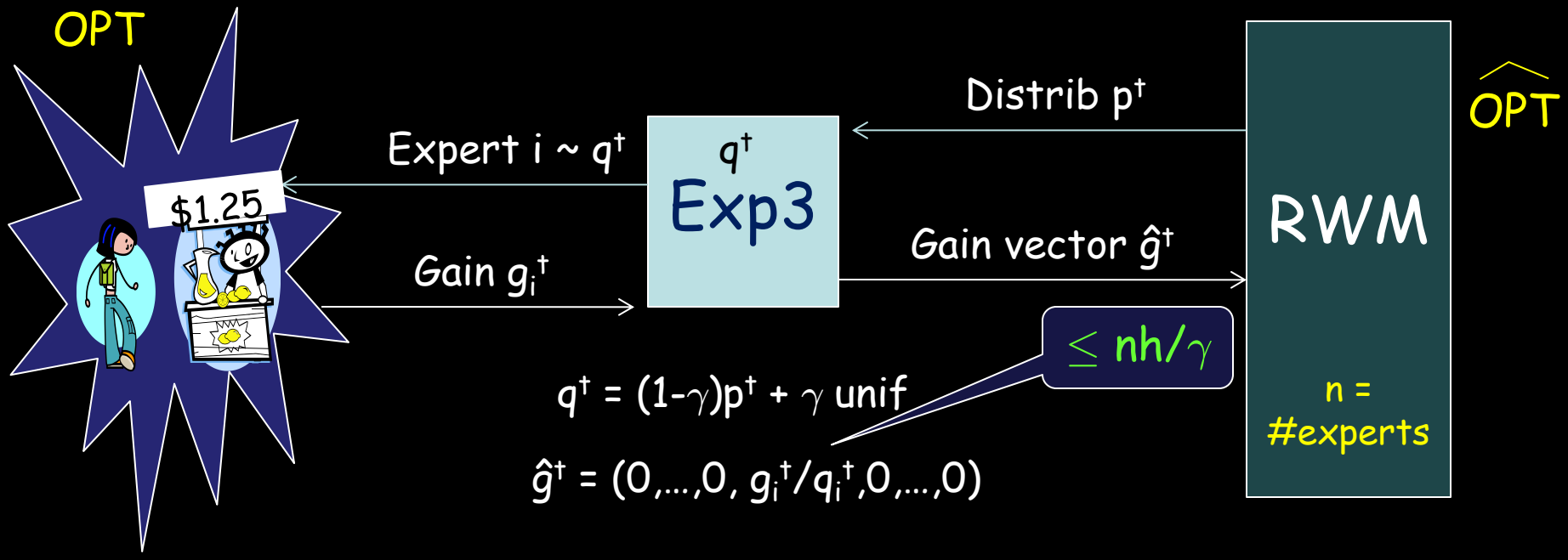
$\hat{g}^t = (0,\ldots,0, g_i^t/q_i^t, 0,\ldots,0)$

1. RWM believes gain is: $p^t \cdot \hat{g}^t = p_i^t(g_i^t/q_i^t) \equiv g^t_{RWM}$
2. $\sum_t g^t_{RWM} \geq \widehat{OPT}\,(1-\epsilon) - O(\epsilon^{-1} nh/\gamma \log n)$
3. Actual gain is: $g_i^t = g^t_{RWM}\,(q_i^t/p_i^t) \geq g^t_{RWM}(1-\gamma)$
4. $E[\widehat{OPT}] \geq OPT$. Because $E[\hat{g}_j^t] = (1- q_j^t)0 + q_j^t(g_j^t/q_j^t) = g_j^t$,
   so $E[\max_j[\sum_t \hat{g}_j^t]] \geq \max_j [\, E[\sum_t \hat{g}_j^t]\,] = OPT$.

Multi-armed bandit problem

Exponential Weights for Exploration and Exploitation (exp^3)

[Auer,Cesa-Bianchi,Freund,Schapire]



Conclusion ($\gamma = \epsilon$):

$E[Exp3] \geq OPT(1-\epsilon)^2 - O(\epsilon^{-2}\, nh \log(n))$

Balancing would give $O((OPT\ nh \log n)^{2/3})$ regret because of ϵ^{-2}. But can reduce to ϵ^{-1} and $O((OPT\ nh \log n)^{1/2})$ with better analysis.

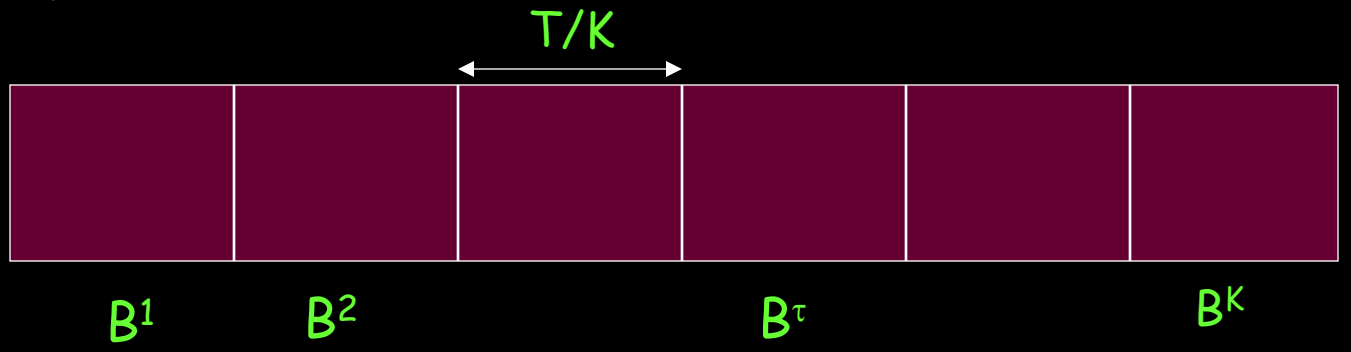
Another reduction (not as good but more generic)

Given: algorithm A for full-info setting with regret $\leq R(T)$.

Goal: use in black-box manner for bandit problem.

Preliminaries:

- First, suppose we break our T time steps into K blocks of size T/K each.



- Use same distrib throughout block and update based on average cost vector c^τ for block τ.
- Then, will get regret $\leq R(K)$ T/K.
- What if we instead update on cost vector $c' \in [0,1]^N$ that's a random variable whose expectation is correct?

Because really paying T/K c^τ per block

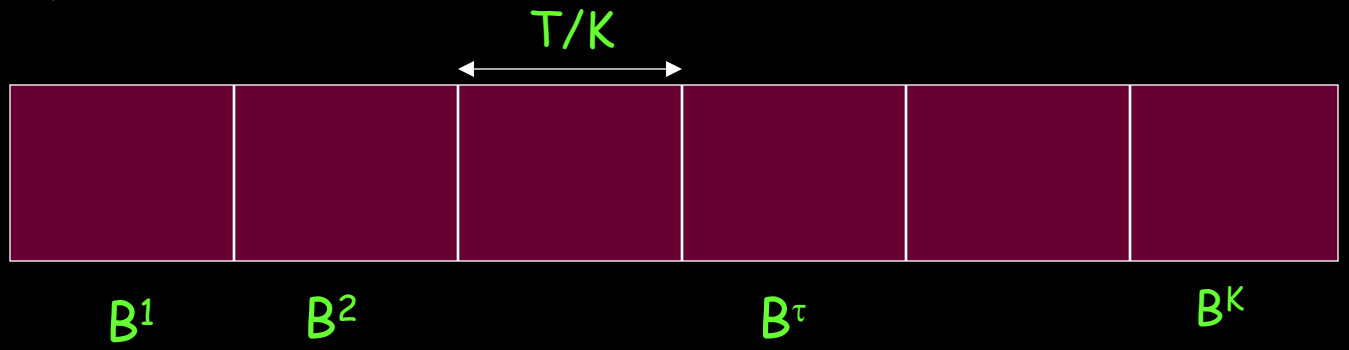
Another reduction (not as good but more generic)

Given: algorithm A for full-info setting with regret $\leq R(T)$.

Goal: use in black-box manner for bandit problem.

Preliminaries:

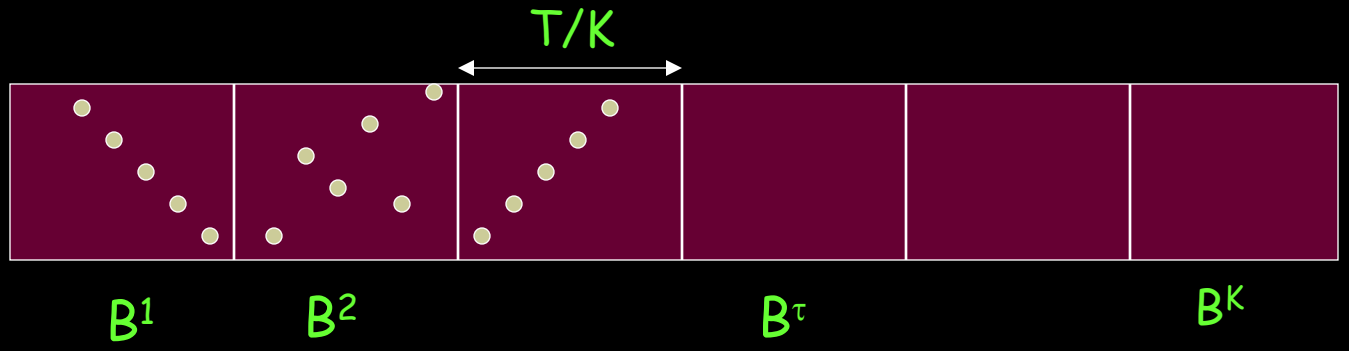
- First, suppose we break our T time steps into K blocks of size T/K each.



- Do at least as well by $\{0,1\} \rightarrow [0,1]$ argument. Still get regret bound R(K) T/K.
- How does this help us for bandit problem?
- What if we instead update on cost vector $c' \in [0,1]^N$ that's a random variable whose expectation is correct?

Experts → Bandit setting

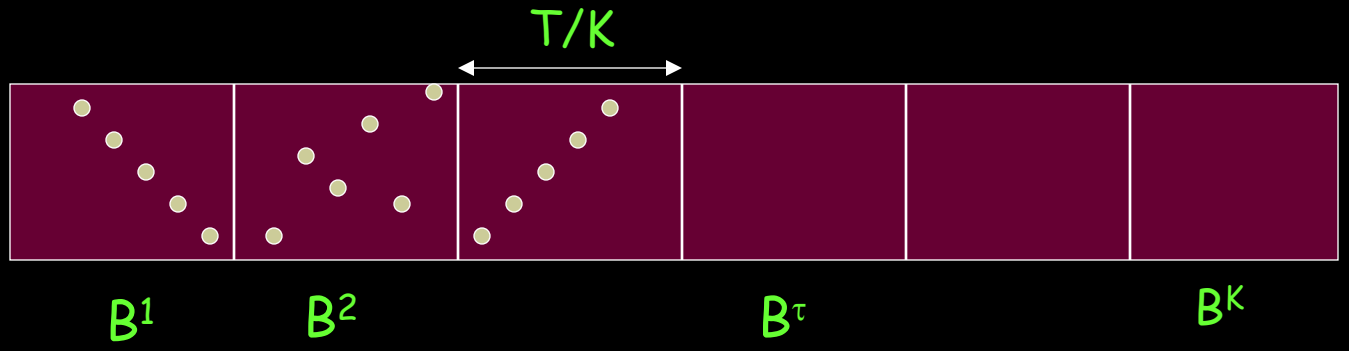
- For bandit problem, for each action, pick random time step in each block to try it as "exploration".
- Define c' only wrt these exploration steps.
- Just have to pay an extra at most NK for cost of this exploration.



- Do at least as well by $\{0,1\} \rightarrow [0,1]$ argument. Still get regret bound R(K) T/K.
- How does this help us for bandit problem?
- What if we instead update on cost vector $c' \in [0,1]^N$ that's a random variable whose expectation is correct?

Experts → Bandit setting

- For bandit problem, for each action, pick random time step in each block to try it as "exploration".
- Define c' only wrt these exploration steps.
- Just have to pay an extra at most NK for cost of this exploration.



- Final bound: R(K) T/K + NK.
- Using $K = (T/N)^{2/3}$ and bound from RWM, get cumulative regret bound of $O(T^{2/3}N^{1/3} \log N)$.

Summary

Can apply algorithms for online decision-making even with very limited feedback.

- Application: which way to drive to work, with only feedback about your own paths; online pricing, even if only have buy/no buy feedback.