Feature-based Dynamic Pricing (Cohen et al., 2020)

AID-554 Project

Pratyush Jha (20410022)
Sudatt Kayal (20118093)

Summary:

- Introduction to the problem setting
- Multi-dimensional binary search (Polytope Pricing algorithm)
- Ellipsoid Pricing algorithm to improve regret w.r.t. dimensions
- Introducing noise to our problem setting
- Shallow Pricing algorithm: a robust version of Ellipsoid Pricing algorithm
- EXP4 Algorithm, and combining it with shallow pricing
- Comparison of our computational results with the ones present in the paper

INTRODUCTION TO THE PROBLEM SETTING

Introduction to contextual bandits setting:

- Consider a multi arm bandit problem with $K \geq 2$ arms.
- At each time t=1,2,… we are getting s1,s2,s3,… from context set S.
- Then the user must learn the best mapping $g: S \rightarrow \{1,\ldots,K\}$ of contexts to arm.
- Application of contextual bandits : news article recommendation.
- An article is suggested to each user : Arm
- The user then either clicks on the article or ignores it : Reward
- If the demographics, past type of content, etc. of user are known, then the article recommendation can be further improved.

Problem Setting:

- An online marketplace is offering **highly differentiated** products.
- The market value of each product is unknown.
- The task is to price the products such that the profit is maximized.
- Features corresponding to each product is provided : Contextual Bandit Problem.
- However, nature selects the features : Adversarial Setting.
- Assumption : The market value of each product is linear in the features of that product.
- A price is set by the seller and the buyer chooses to either buy it or not.
- Based on the past history, the future pricing is done.
- Application : Airbnb, eBay, Banks, etc.

Problems that the paper addresses:

- Many algorithms in contextual bandit settings assume the payoffs to be linear in feature.
- In the present paper, the payoffs are discontinuous pricing functions.
- Multiple papers address the setting of contextual bandits, but with features stochastically obtained.
- In the present paper, the features are obtained adversarially.
- Dynamic pricing problems in high dimensions has been solved in different papers, however, they don’t consider contextual setting.
- Direction of cuts can’t be chosen by us and is decided by the nature adversarially.

ALGORITHMS DISCUSSED IN THE PAPER

One-dimension binary search

- $d = 1, \theta \in [0,1], x_t = 1$ (any value won't matter)
- Initially : $K_1 \in [0,1]$
- Algorithm:
  - $K_t \in [l_t, u_t]$
  - Set $p_t = \frac{1}{2}(u_t + l_t)$
  - If user accepts :
    - Reward $= p_t$
    - $l_t = p_t$
  - If user rejects:
    - Reward $= 0$
    - $u_t = p_t$
  - Do this until $u_t - l_t \leq \epsilon$
    - After this set $p_t = l_t$

Regret Analysis:

- No of explore steps $= \log_2(\frac{1}{\epsilon})$
- Regret $\leq \log_2(\frac{1}{\epsilon}) + (T - \log_2(\frac{1}{\epsilon})) \cdot \epsilon$
- Regret $\leq O(\log_2 T)$ $for\ \epsilon = \frac{1}{T}$

Problem in High Dimensions

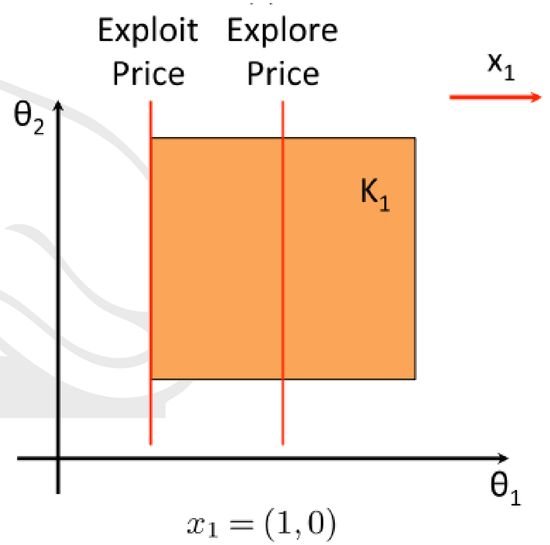
- The value of x_t affects the algorithm now
- Explore then exploit won't work here
- Features selected by nature would never offer an opportunity to learn θ precisely.

$$\text{Eg :} \begin{pmatrix} x_1^{(1)} \\ x_1^{(2)} \\ 0 \\ 0 \end{pmatrix}$$

- Correlation between features by nature
- Some features may appear with non-zero value close to horizon T

Polytope Pricing Algorithm

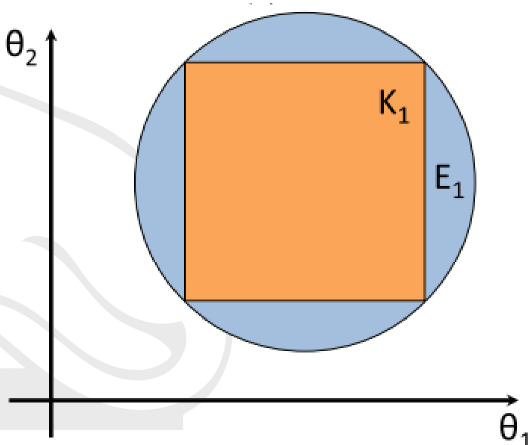
- Exploitation based on amount of information gathered for the particular set of features
- We define : $\underline{b_t} = \min_{\hat{\theta} \in K_t} \hat{\theta}' x_t$ and $\overline{b_t} = \max_{\hat{\theta} \in K_t} \hat{\theta}' x_t$
- $p_t = \underline{b_t}$ is the exploit price
- $p_t = \frac{1}{2}(\underline{b_t} + \overline{b_t})$ is the explore price (provides most information of the buyer's valuation)
- Algorithm :
  - Compute $\underline{b_t}$ and $\overline{b_t}$
  - If $\overline{b_t}$ - $\underline{b_t} \leq \epsilon$: Set exploit price $p_t = \underline{b_t}$
  - If $\overline{b_t}$ - $\underline{b_t} > \epsilon$: Set explore price $p_t = \frac{1}{2}(\underline{b_t} + \overline{b_t})$
    - If sale occurs : $K_{t+1} = K_t \cap \{\theta \in R^d: \theta' x_t \geq p_t\}$
    - If no sale : $K_{t+1} = K_t \cap \{\theta \in R^d: \theta' x_t \leq p_t\}$



- Worst case regret : $\Omega(Ra^d \ln(T))$, for some constant $a > 1$.

Ellipsoid Pricing Algorithm

- Inspired by Khachiyan's (1979) ellipsoid method
- Instead of uncertainty K_t , round it up to smallest ellipsoid E_t containing K_t
- E_t: Lowner-John ellipsoid of the set K_t
- The bounds are now defined as : $\underline{\mathrm{b}_\mathrm{t}} = \min_{\hat{\theta} \in E_t} \hat{\theta}' x_t$ and $\overline{\mathrm{b}_\mathrm{t}} = \max_{\hat{\theta} \in E_t} \hat{\theta}' x_t$
- Algorithm :
  - Compute $\underline{\mathrm{b}_\mathrm{t}}$ and $\overline{\mathrm{b}_\mathrm{t}}$
  - If $\overline{\mathrm{b}_\mathrm{t}}$ - $\underline{\mathrm{b}_\mathrm{t}} \leq \epsilon$: Set exploit price $\mathrm{p}_\mathrm{t} = \underline{\mathrm{b}_\mathrm{t}}$
  - If $\overline{\mathrm{b}_\mathrm{t}}$ - $\underline{\mathrm{b}_\mathrm{t}} > \epsilon$: Set explore price $\mathrm{p}_\mathrm{t} = \frac{1}{2}(\underline{\mathrm{b}_\mathrm{t}} + \overline{\mathrm{b}_\mathrm{t}})$
    - If sale occurs : $H_{t+1} = E_t \cap \{\theta \in \mathrm{R}^\mathrm{d}: \theta' x_t \geq p_t\}$
    - If no sale :$H_{t+1} = E_t \cap \{\theta \in \mathrm{R}^\mathrm{d}: \theta' x_t \leq p_t\}$
  - E_t is the Lowner-John ellipsoid of half ellipsoid H_{t+1}



Primer on ellipsoids

- A symmetric matrix is said to be positive definite if all its eigenvalues are positive.
- An ellipsoid E is defined by a vector(called as the center) $a \in R^d$, and a positive definite dxd matrix A, as:

$$E(A,a) := \{\theta \in \mathbb{R}^d : (\theta - a)' A^{-1} (\theta - a) \leq 1\}$$

- $\underset{\theta \in E(A,a)}{\operatorname{argmax}} \theta' x = a + b$ and $\underset{\theta \in E(A,a)}{\operatorname{argmin}} \theta' x = a - b$, where $b = \frac{Ax}{\sqrt{x' A x}}$
- The hyperplane perpendicular to x passing through a is given by $x'(\theta - a)$, which cuts the ellipsoid into two symmetric pieces.
- The smallest ellipsoid containing $E(A,a) \cap \{\theta \in R^d : x'(\theta - a) \leq 0\}$ is $E\left(\tilde{A}, a - \frac{1}{d+1} b\right)$ and $E(A,a) \cap \{\theta \in R^d : x'(\theta - a) \geq 0\}$ is $E\left(\tilde{A}, a + \frac{1}{d+1} b\right)$, where

$$\tilde{A} = \frac{d^2}{d^2 - 1}\left(A - \frac{2}{d+1} bb'\right)$$

Advantages of EllipsoidPricing Algorithm

- Thus we have $\underline{\mathrm{b}_\mathrm{t}} = \min_{\hat{\theta}\in E_t} \hat{\theta}' x_t = x_t' \mathrm{a}_\mathrm{t} - \sqrt{x' A x}$ and $\overline{\mathrm{b}_\mathrm{t}} = x_t' \mathrm{a}_\mathrm{t} + \sqrt{x' A x}$
- Hence matrix product calculation replaces linear programming, hence more efficient computationally
- The algorithm only needs to keep track of dxd matrix and d-dimensional vector
- The worst case regret of EllipsoidPricing algorithm with parameter $\epsilon = \frac{\mathrm{Rd}^2}{\mathrm{T}}$ is $O\left(Rd^2 \ln\left(\frac{T}{d}\right)\right)$
- Enlarging uncertainty is leading to regularization.

Noisy Valuations

- Now we allow for additive noise in valuation $v_t = \theta' x + \delta_t$, where δ_t is i.i.d zero-mean random variable representing an error in estimate of v_t.
- We assume that the distribution of the noise δ_t is fixed over time, known, and σ-sub-Gaussian.
- The focus then shifts towards learning weights rather than the noise distribution itself.
- A distribution is σ-sub-Gaussian if $\Pr(|\delta_t| > \delta) \geq e^{-\frac{t^2}{2\sigma^2}}$ for all t>0.
- The author proposes two algorithms : ShallowPricing and EllipsoidEXP4.
- ShallowPricing is a robust version of EllipsoidPricing for low noise setting.
- EllipsoidEXP4 combines ShallowPricing with EXP4(an algorithm for contextual bandit), which proves to be the most efficient in noise setting.

SHALLOW PRICING Algorithm:

A robust version of Ellipsoid Pricing

- We define a parameter δ as:

$$\delta = \sqrt{2\sigma \ln T}$$

- Since δ_t is assumed to be a σ-sub-Gaussian :

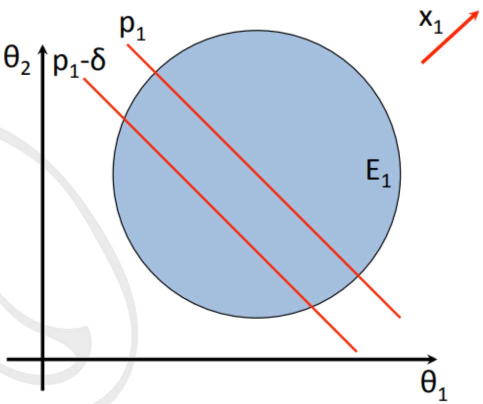
$$\Pr(|\delta_t| \leq \delta \text{ for all } t = 1, \ldots, T) \geq 1 - Te^{-\ln^2 T} \geq 1 - 1/T,$$

where the last inequality holds for $T \geq 8$. Therefore, the noise terms δ_t are bounded by δ in absolute value.

- Hence, on choosing a price p_t, we now infer: $\theta' v_t \geq p_t - \delta$ (instead of $\theta' v_t \geq p_t$)

We remove elements from the set as if we had used the price, $p_t + \delta$, i.e.,

$$K_{t+1} = K_t \cap \{\theta \in R^d : \theta' x_t \leq p_t + \delta\}$$



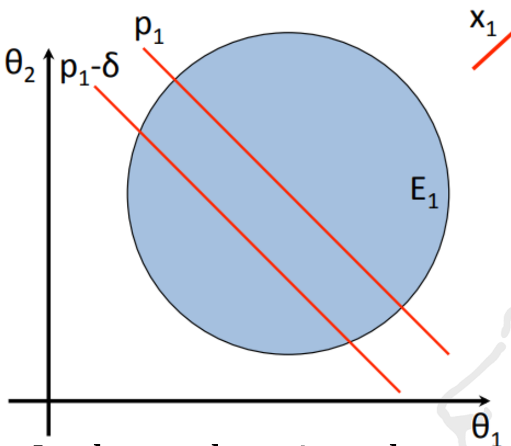
- Similarly, if no sale occurs, we infer: $\theta' v_t \leq p_t + \delta$ (instead of $\theta' v_t \leq p_t$)

We remove elements from the set as if we had used the price $p_t + \delta$, i.e.,

$$K_t + 1 = K_t \cap \{\theta \in \mathbf{R}^d : \theta' x_t \leq p_t + \delta\}$$

SHALLOW PRICING Algorithm is the Ellipsoid Pricing algorithm with a buffer parameter (δ) to account for the σ-sub-gaussian noise term (δ_t) in the market value of the product.

SHALLOW PRICING Algorithm (Contd.)



- The depth of a cut is given by: $\alpha_t = -\dfrac{\delta}{\sqrt{x_t' A_t x_t}}$, $\alpha_t \in [-1, 0]$
- The case, $\alpha_t = 0$ represents a central cut of the ellipsoid.
- The analysis does not include the case of positive depth.

In the exploration phase:

- If $\theta' v_t \geq p_t - \delta$: $E(A_{t+1}, a_t - \frac{1+d\alpha_t}{d+1} b_t)$
- Else: $E(A_{t+1}, a_t + \frac{1+d\alpha_t}{d+1} b_t)$
- In exploitation phase: $p_t = \underline{b}_t - \delta$

A_{t+1} and b_t are given as:

$$A_{t+1} = \frac{d^2}{d^2-1}(1-\alpha_t^2)\left(A_t - \frac{2(1+d\alpha_t)}{(d+1)(1+\alpha_t)} b_t b_t'\right)$$

$$b_t = A_t x_t / \sqrt{x_t' A_t x_t}$$

Results:

1. By using $\epsilon = \max\{Rd^2/T, 4d\delta\}$, we obtain $N = O(d^2 \ln(\min\{T/d, R/\delta\}))$
2. SHALLOW PRICING is robust for $\sigma \sim O(1/T \ln T)$
3. Regret of SHALLOW PRICING = $O(Rd^2 \ln(T/d))$

EXP4 Algorithm:
The generic setup of EXP4

- The generic setup of EXP4 comprises a space **X** of contexts, a space **A** of actions, and a set Π of policies.

$$\pi \in \Pi: \qquad \pi(x_t) : \mathbf{X} \rightarrow \mathbf{A}, \qquad x_t \in \mathbf{X}, a \in \mathbf{A}$$

- The learner chooses an action at and obtains the reward $r_t(a_t)$.

- The regret is defined as:

$$\text{REGRET} = \mathbb{E}\left[\max_{\pi \in \Pi} \sum_t r_t(\pi(x_t)) - \sum_t r_t(a_t)\right]$$

- The EXP4 algorithm maintains weights $w_t(\pi)$ for each policy $\pi \in \Pi$, which are initialized as $w_1(\pi) = 1$ for all policies.

EXP4 Algorithm:
The algorithm explained

1. For each t, a policy π is drawn with probability proportional to $w_t(\pi)$.
2. According to the recommendation by the chosen policy, we get, $a_t = \pi(x_t)$.
3. Based on the action, we observe the reward $r_t(a_t)$.
4. Calculate:

$$\tilde{r}_t(\pi) = \begin{cases} r_t(a_t) \cdot \dfrac{\sum_{\pi \in \Pi} w(\pi)}{\sum_{\pi:\ \pi(x_t)=a_t} w(\pi)} & \text{if } \pi(x_t) = a_t; \\ 0 & \text{otherwise,} \end{cases}$$

5. Update the weights of the policies according to the given η:

$$w_{t+1}(\pi) = w_t(\pi) \cdot \exp(\eta \cdot \tilde{r}_t(\pi))$$

6. The parameter η (learning rate) is a constant when the algorithm is executed. The performance of the algorithm is sensitive with respect to the value of η taken.

EXP4 Algorithm:

Exp4 (Exponential weights algorithm for Exploration and Exploitation with Experts) without mixing:

Parameter: a non-increasing sequence of real numbers $(\eta_t)_{t \in \mathbb{N}}$.

Let q_1 be the uniform distribution over $\{1, \ldots, N\}$.

For each round $t = 1, 2, \ldots, n$

(1) Get expert advice $\xi_t^1, \ldots, \xi_t^N$, where each ξ_t^j is a probability distribution over arms.

(2) Draw an arm I_t from the probability distribution $p_t = (p_{1,t}, \ldots, p_{K,t})$, where $p_{i,t} = \mathbb{E}_{j \sim q_t} \xi_{i,t}^j$.

(3) Compute the estimated loss for each arm

$$\widetilde{\ell}_{i,t} = \frac{\ell_{i,t}}{p_{i,t}} \mathbb{1}_{I_t = i} \qquad i = 1, \ldots, K \ .$$

(4) Compute the estimated loss for each expert

$$\widetilde{y}_{j,t} = \mathbb{E}_{i \sim \xi_t^j} \widetilde{\ell}_{i,t} \qquad j = 1, \ldots, N \ .$$

(5) Update the estimated cumulative loss for each expert $\widetilde{Y}_{j,t} = \sum_{s=1}^{t} \widetilde{y}_{j,s}$ for $j = 1, \ldots, N$.

(6) Compute the new probability distribution over the experts $q_{t+1} = (q_{1,t+1}, \ldots, q_{N,t+1})$, where

$$q_{j,t+1} = \frac{\exp\left(-\eta_t \widetilde{Y}_{j,t}\right)}{\sum_{k=1}^{N} \exp\left(-\eta_t \widetilde{Y}_{k,t}\right)} \ .$$

Source: S´ebastien Bubeck1 and Nicol`o Cesa-Bianchi (2012)

EXP4 Algorithm:
Instantiating in the noiseless regime

For a fixed discretization parameter $\gamma \geq 0$,

Discretization operator ($\lfloor . \rfloor_\gamma$) :

- For $y \in \mathbf{R}$:

$$\lfloor y \rfloor_\gamma = \gamma \cdot \lfloor y/\gamma \rfloor$$

- For a vector $y \in \mathbf{R}^d$:

$$(\lfloor y \rfloor_\gamma)_i = \lfloor y_i \rfloor_\gamma$$

- For a set $K \subseteq \mathbf{R}^d$:

$$\lfloor K \rfloor \gamma = \{\lfloor y \rfloor \gamma : y \in K\}$$

Assume: $K_1 = [0, 1]^d$

Associate a policy with each vector $\theta \in \lfloor [0, 1]^d \rfloor_\gamma$: $|\Pi| = O\,(1/\gamma^d)$

Each policy π_θ associate x_t with the following price $\pi_\theta(x_t) = \lfloor \theta' x_t \rfloor_{\gamma \sqrt{d}} - \gamma \sqrt{d}$: $|\mathbf{A}| \leq O((\gamma \sqrt{d})^{-1})$

Regret is of the order of $\tilde{O}\left(T^{2/3} d^{1/3}\right)$.

EXP4 Algorithm:
Instantiating in the noisy regime

Define:

$$p^*(v) = \arg\max_p p \cdot \Pr(v + \delta \geq p)$$

For every discretized θ, re-define the policy to be:

$$\pi_\theta(x_t) = p^* \left(\lfloor \theta' x_t \rfloor_{\gamma\sqrt{d}} - \gamma\sqrt{d} \right)$$

Regret of the order of $\tilde{O}\left(T^{2/3} d^{1/3}\right)$ still holds in the noisy regime as well.*

* According to our paper

ELLIPSOID EXP4:

Combining SHALLOW PRICING and EXP4 Algorithm

Under a high-noise setting (e.g., $\sigma = O(1)$), the performance of SHALLOW PRICING deteriorates since adding a buffer term to all exploit prices becomes too costly.

Hence, the algorithm has to allow some learning during the exploitation periods as well.

ELLIPSOID EXP4 Algorithm:

- In the exploration phase, we use **SHALLOW PRICING:**
  - The price as $p_t = (b^{min}{}_t + b^{max}{}_t)/2$
  - Update the uncertainty set according to the Shallow Pricing rule.
  - We restart the EXP4 algorithm by resetting to 1 the weights of all policies in $\lfloor \gamma(K_t+1) \rfloor_\gamma$.

- In the exploitation phase, we use **EXP4:**
  - EXP4 algorithm in the noisy regime with parameter η.
  - The uncertainty set is unchanged, i.e., $K_{t+1} = K_t$.

Regret Analysis of ELLIPSOID EXP4:

- In low-noise situations:

$$\sigma = O(1/T \ln T)\colon O\left(d^{5/2}\ln(T/d)\cdot\left[1+T^{2/3}d^{2/3}(\sigma\ln(T))^{1/3}\sqrt{\ln(T/\sigma)}\right]\right)$$

- In medium-noise situations:

$$\sigma = O(1) \quad : \tilde{O}(T^{2/3})$$

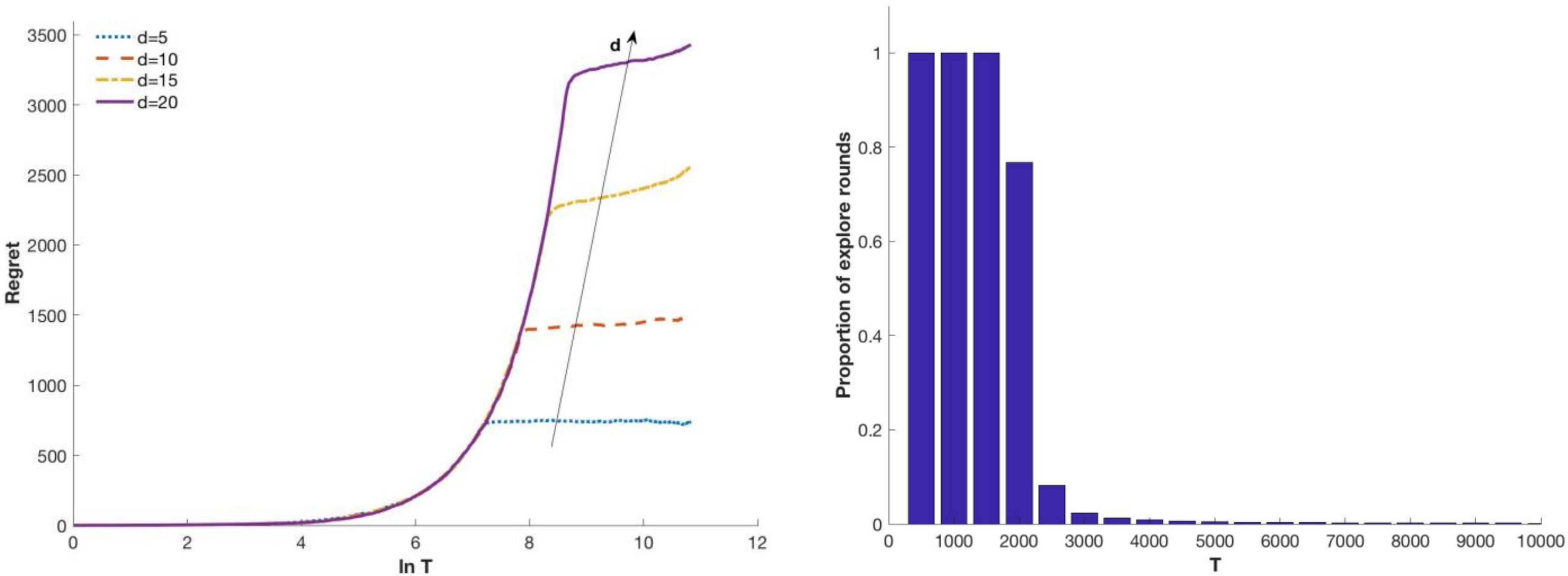
- In high-noise situations:

$$\sigma = O(1/\sqrt{T}) : \tilde{O}(\sqrt{T})$$

$$\sigma = O(T^{-2/3}) : \tilde{O}(T^{4/9})$$

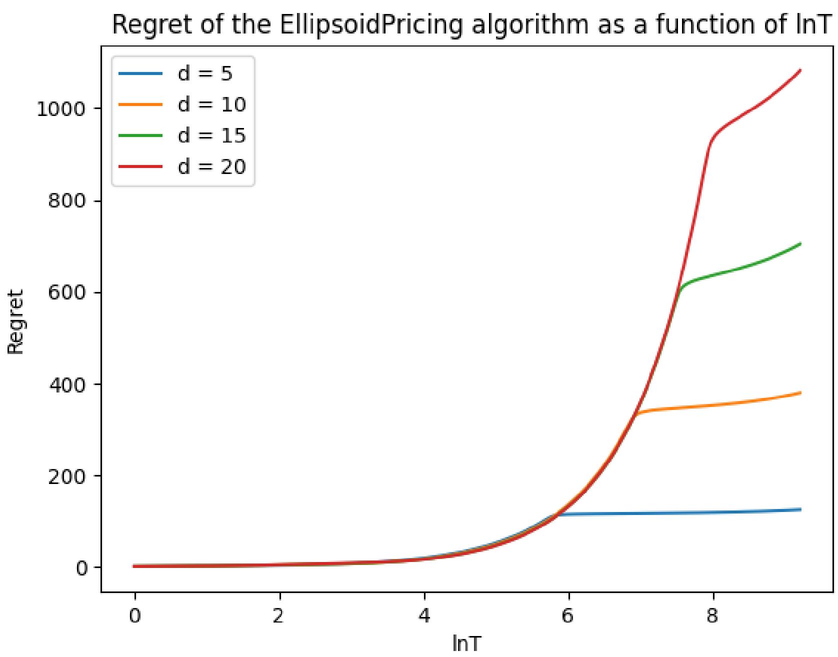
RESULTS

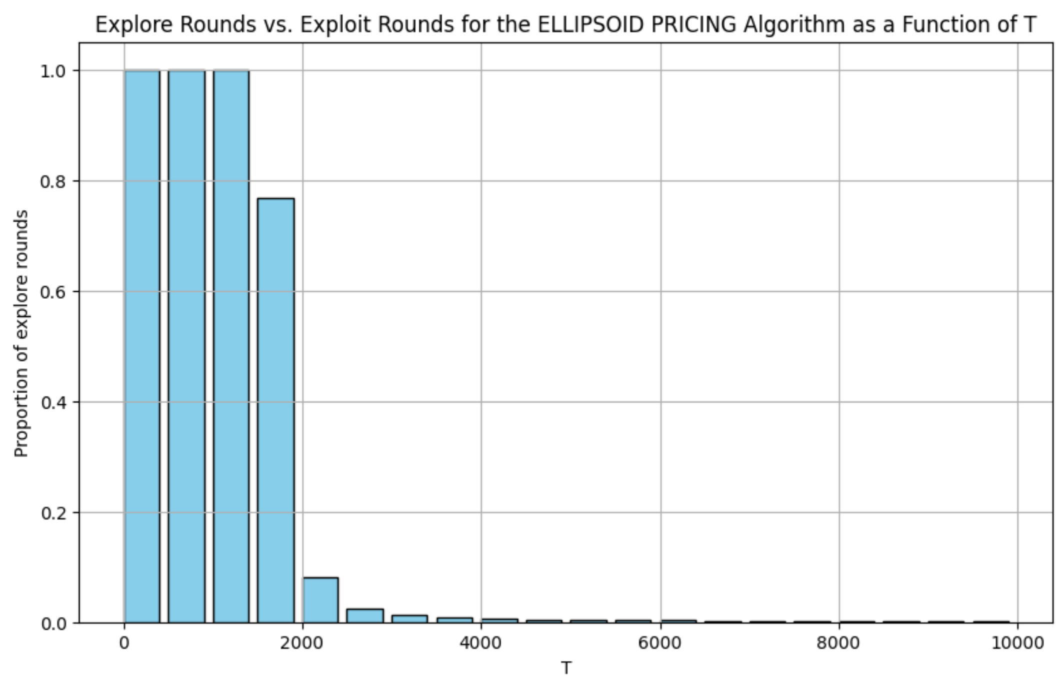
ELLIPSOID PRICING:
Regret analysis plots (Cohen et al., 2020)



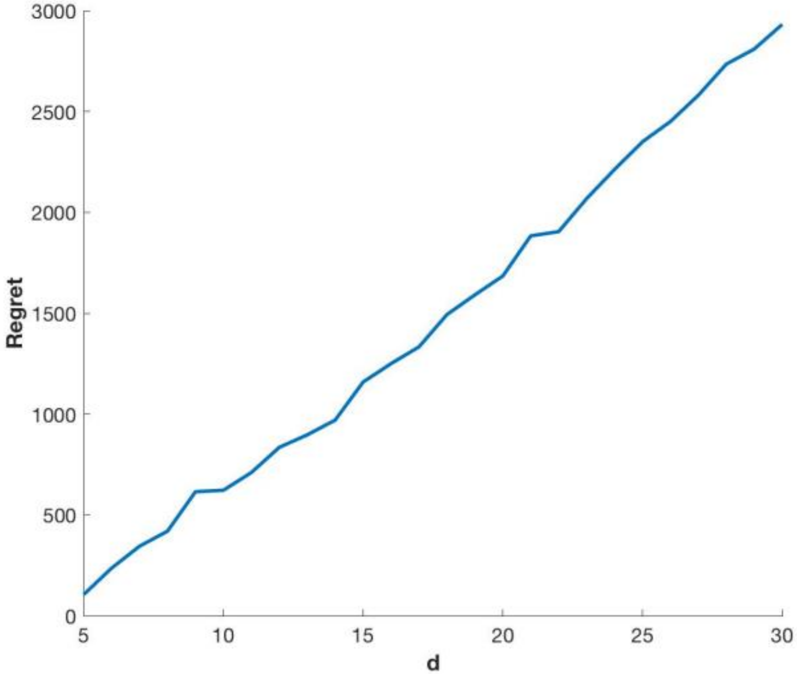
(a) $d \in \{5, 10, 15, 20\}$

ELLIPSOID PRICING:
Regret analysis plots (our work)



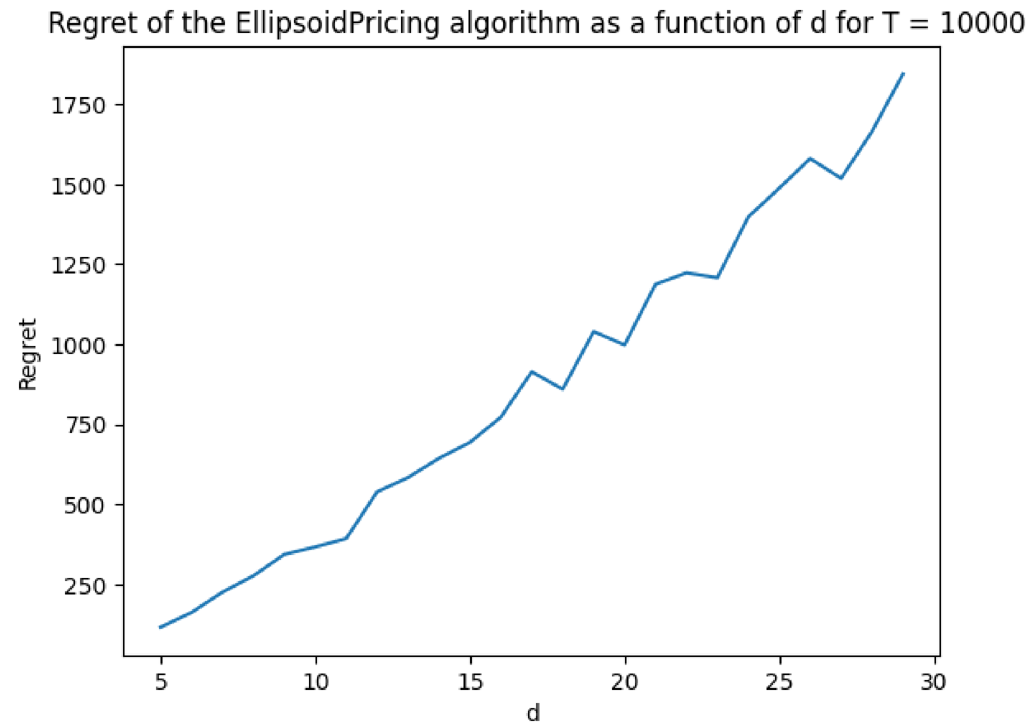


ELLIPSOID PRICING:
Regret vs dimensions plot (Cohen et al., 2020 vs our code)



(b) $T = 10,000$

Cohen et al., 2020

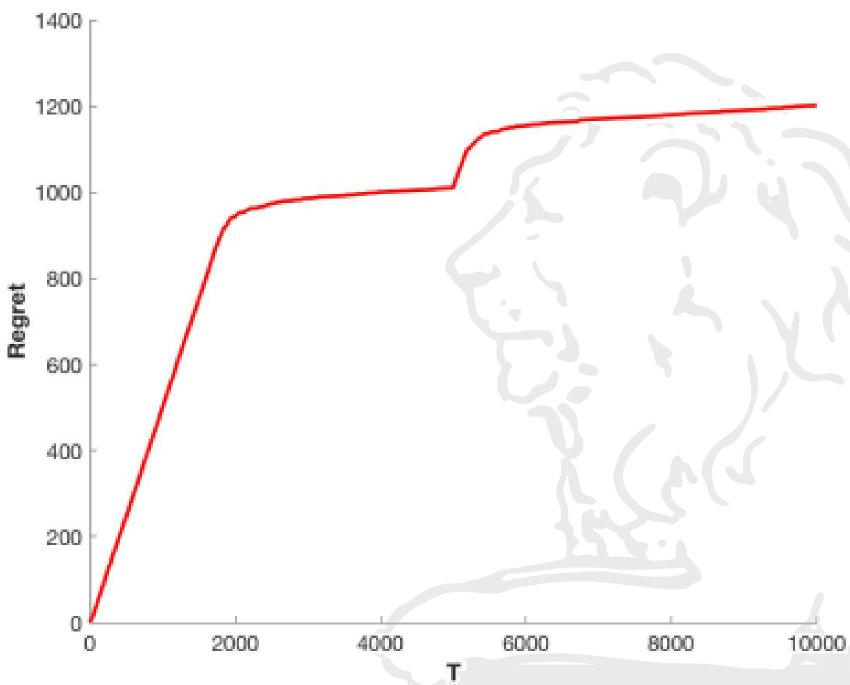


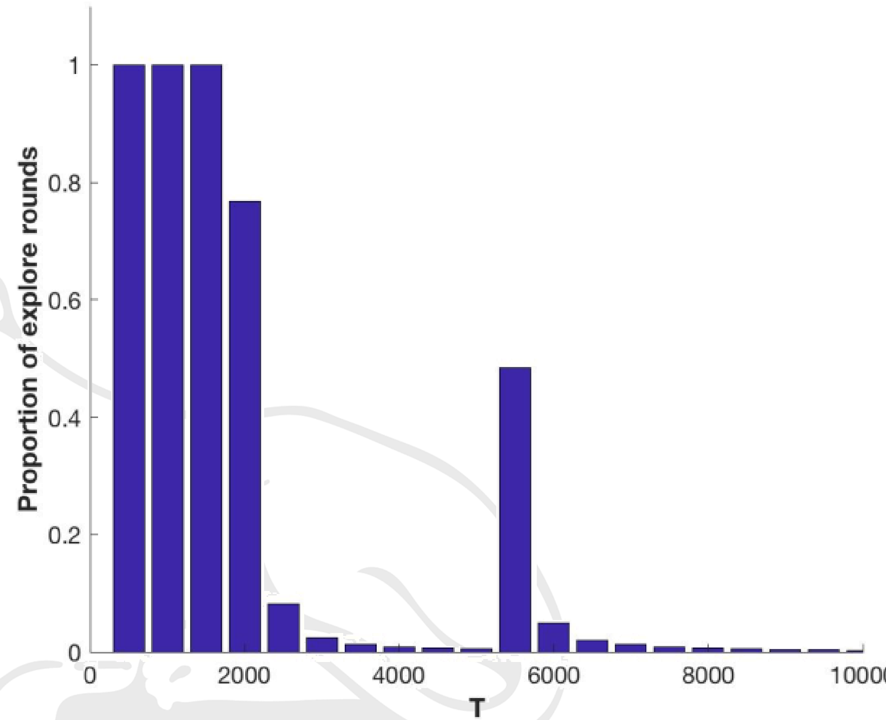
Our code

Observation: The graph shows linear dependency!

ELLIPSOID PRICING: ADAPTABILITY to changed distributions of x_t:

Normal then uniform features (Cohen et al., 2020)

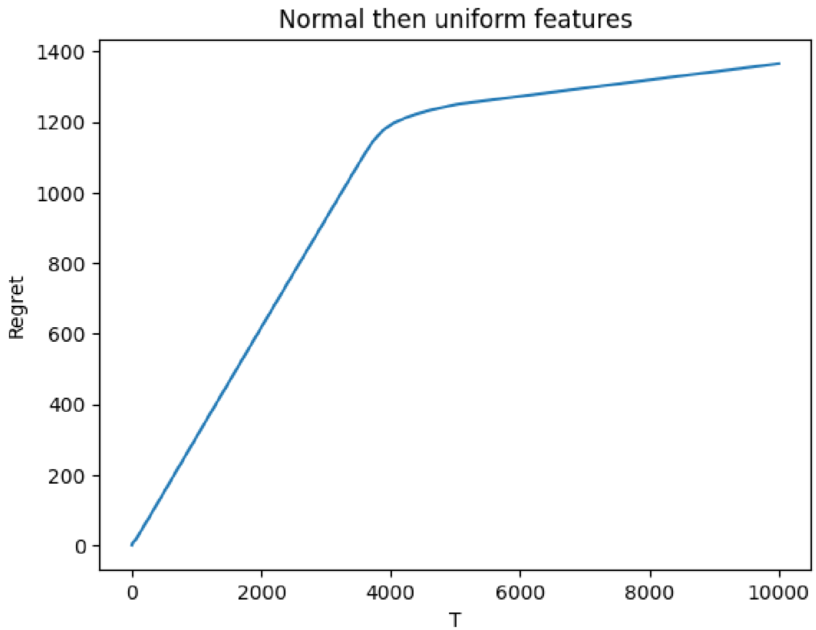


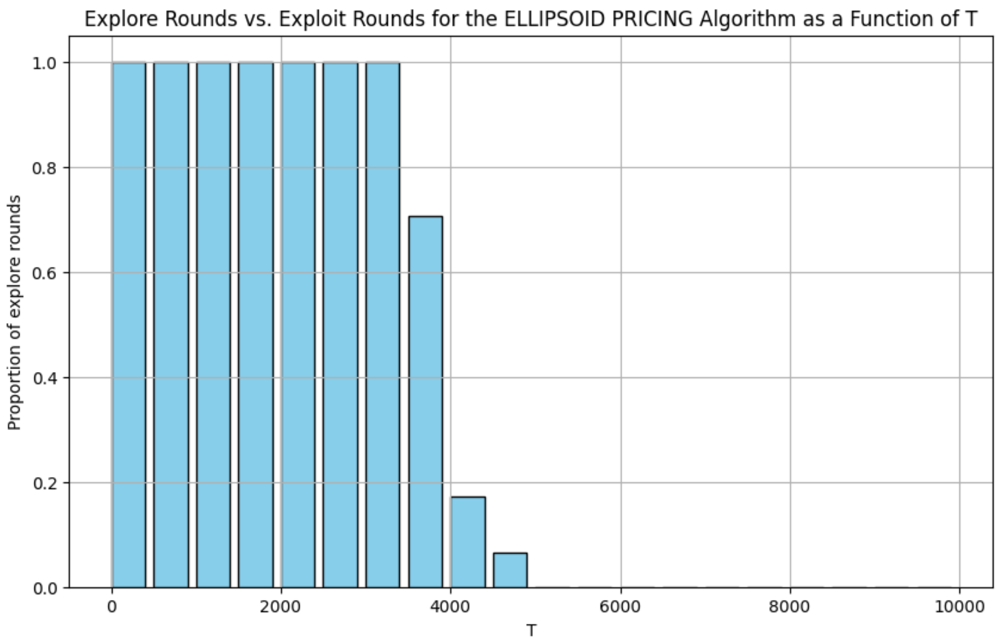


(a) Normal then uniform features.

ELLIPSOID PRICING: ADAPTABILITY to changed distributions of x_t:

Normal then uniform features (our work)

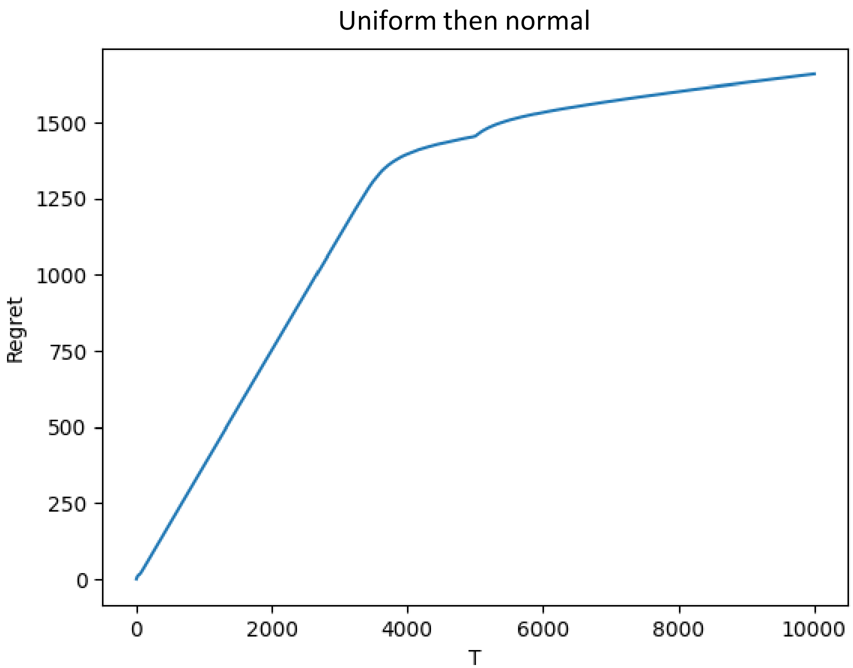


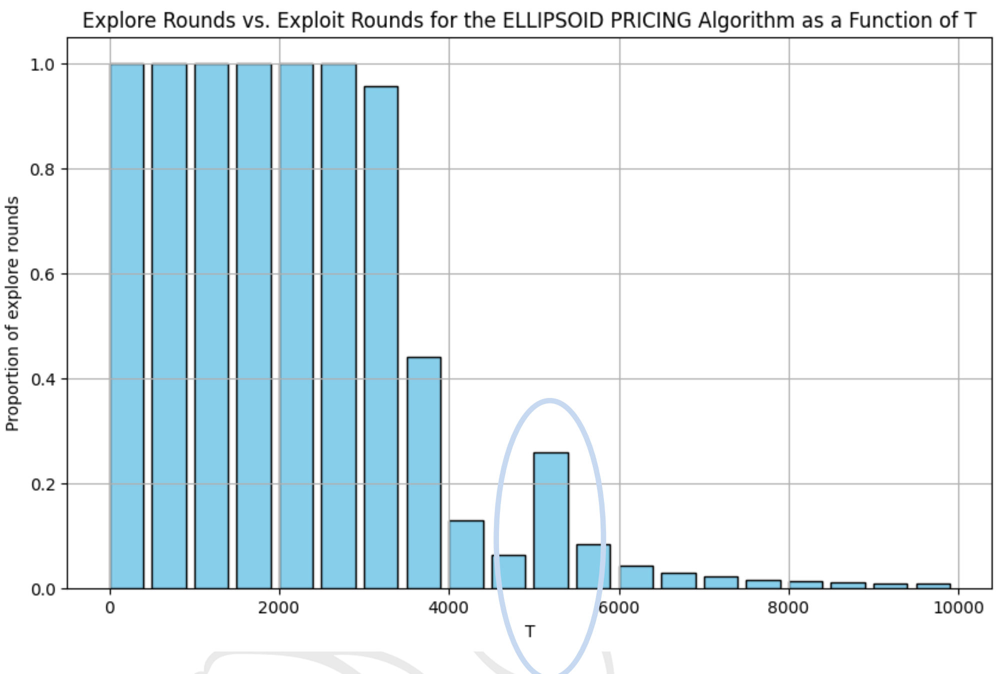


Regret of the EllipsoidPricing algorithm as a function of T for d = 15 when the distribution of the features changes at T /2.

ELLIPSOID PRICING: ADAPTABILITY to changed distributions of x_t:

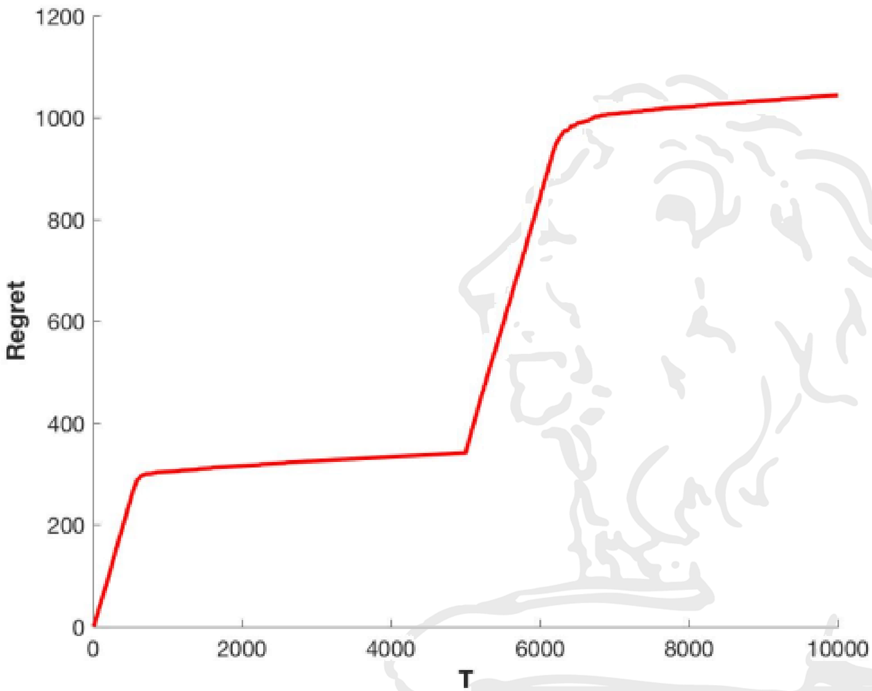
Uniform then normal features(our work)



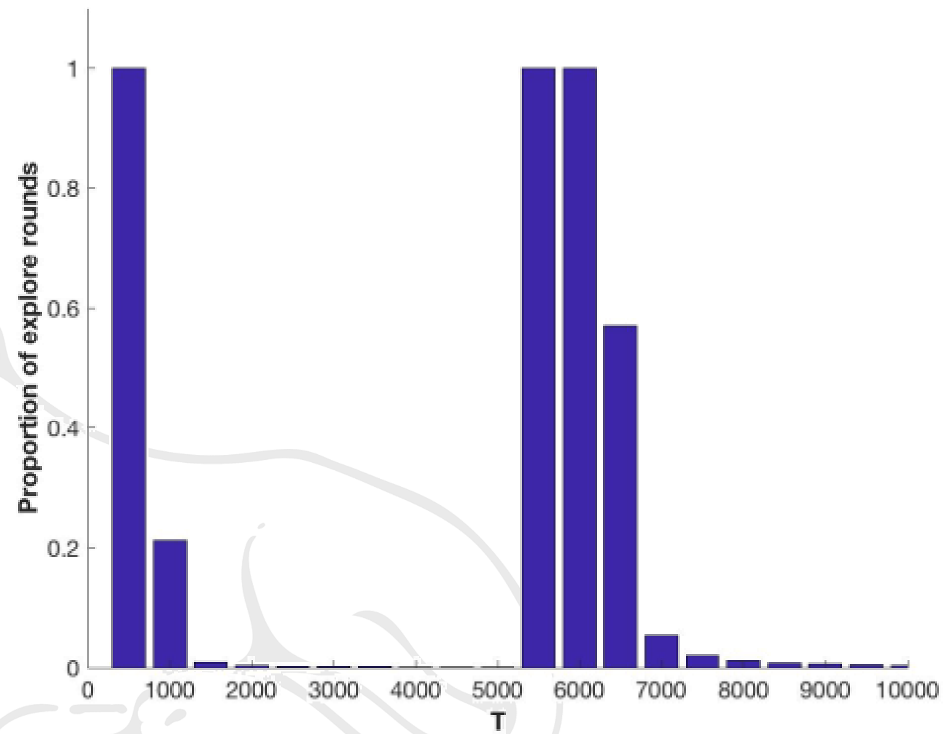


Regret of the EllipsoidPricing algorithm as a function of T for d = 15 when the distribution of the features changes at T /2.

ELLIPSOID PRICING: ADAPTABILITY to changed distributions of x_t:
Half then full features (Cohen et al., 2020)



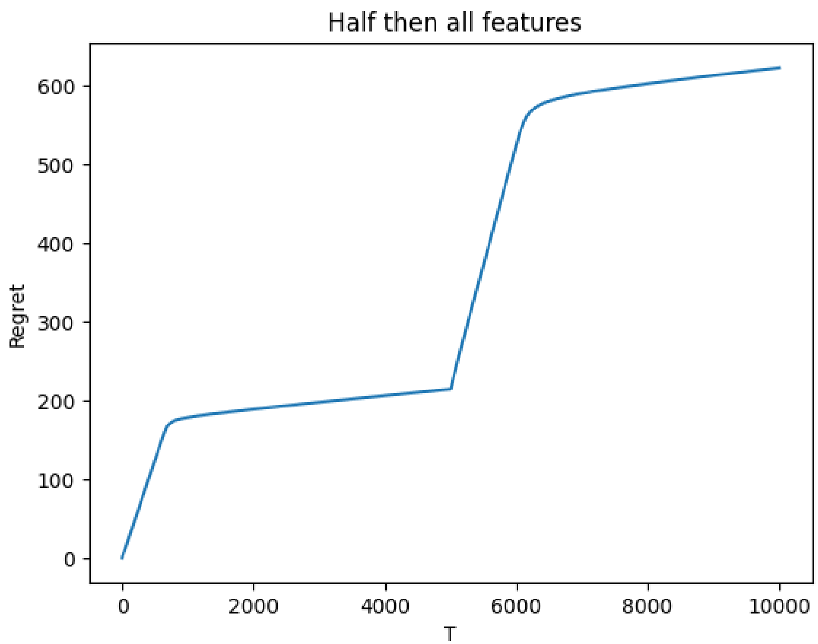
(b) Half then all features.

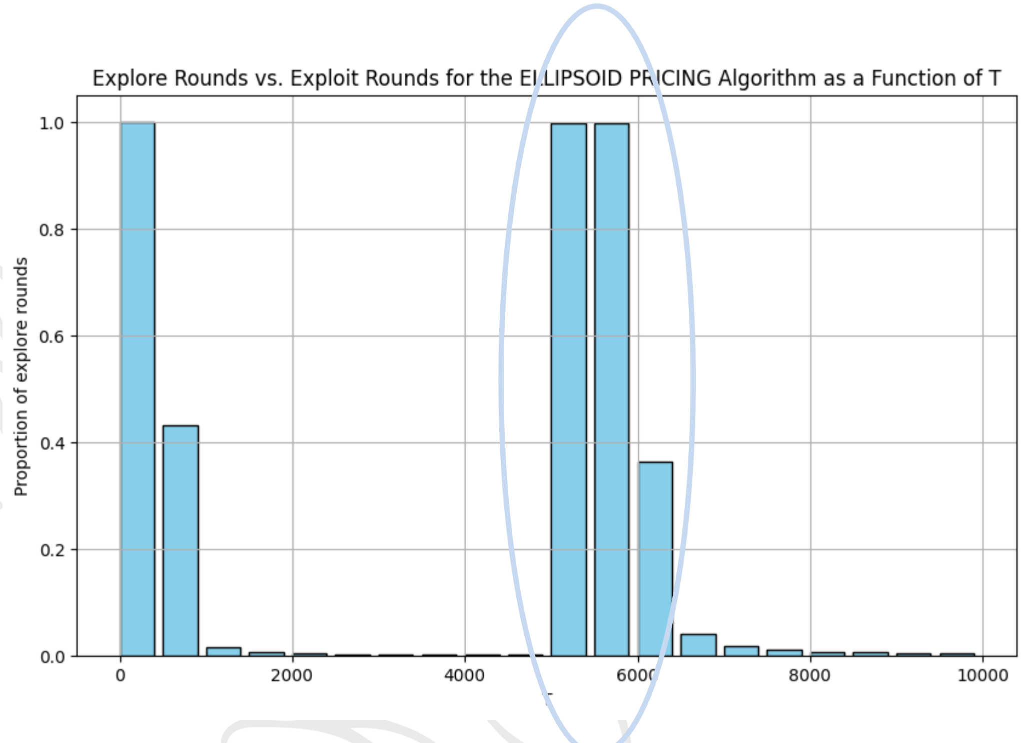


(b) Half then all features.

ELLIPSOID PRICING: ADAPTABILITY to changed distributions of x_t:

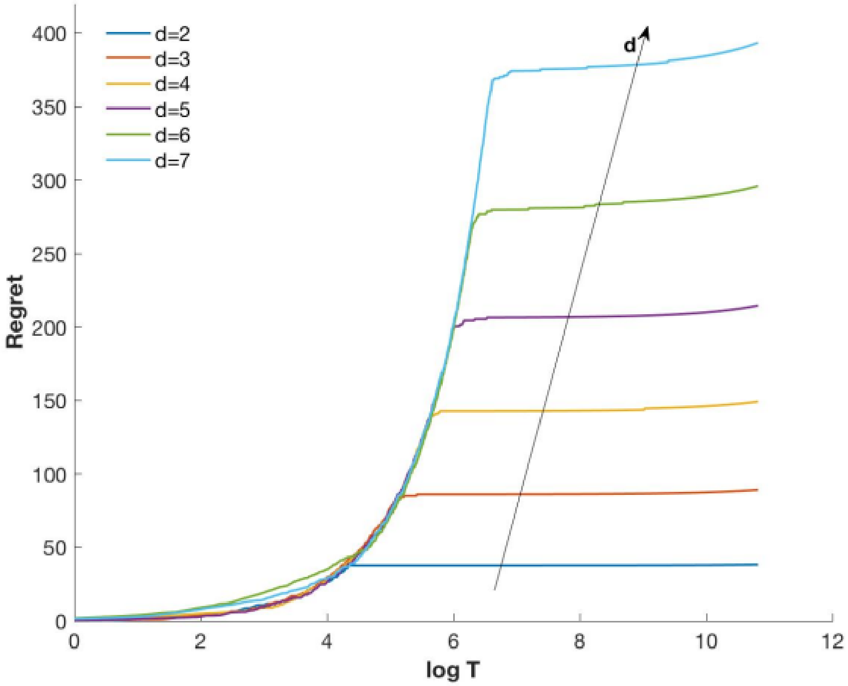
Half then full features (our work)



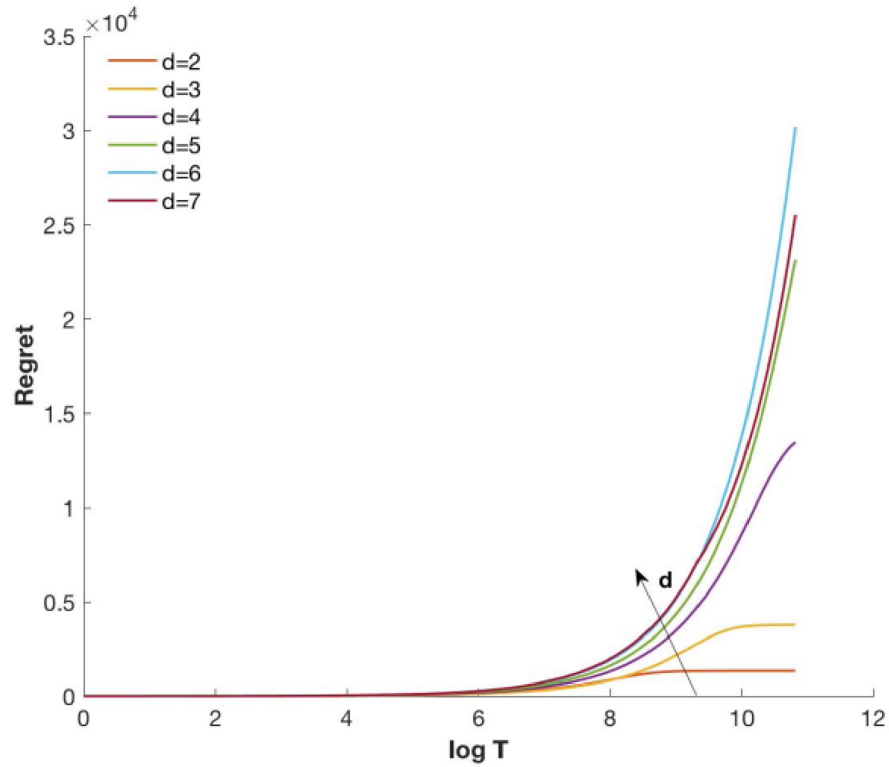


Regret of the EllipsoidPricing algorithm as a function of T for d = 15 when the distribution of the features changes at T /2.

ELLIPSOID PRICING vs EXP4 Algorithm (Cohen et al., 2020):



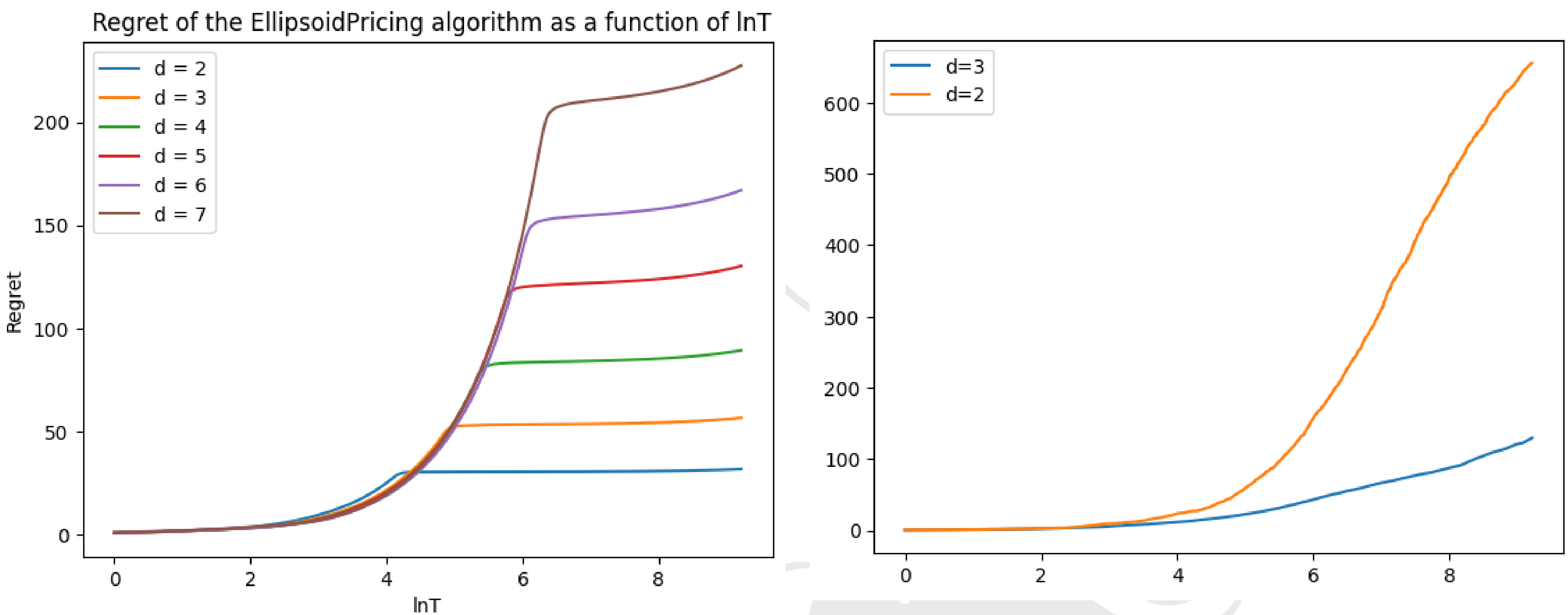
(a) Regret of EllipsoidPricing.



(b) Regret of EXP4 (the y-axis is scaled by 10^4).

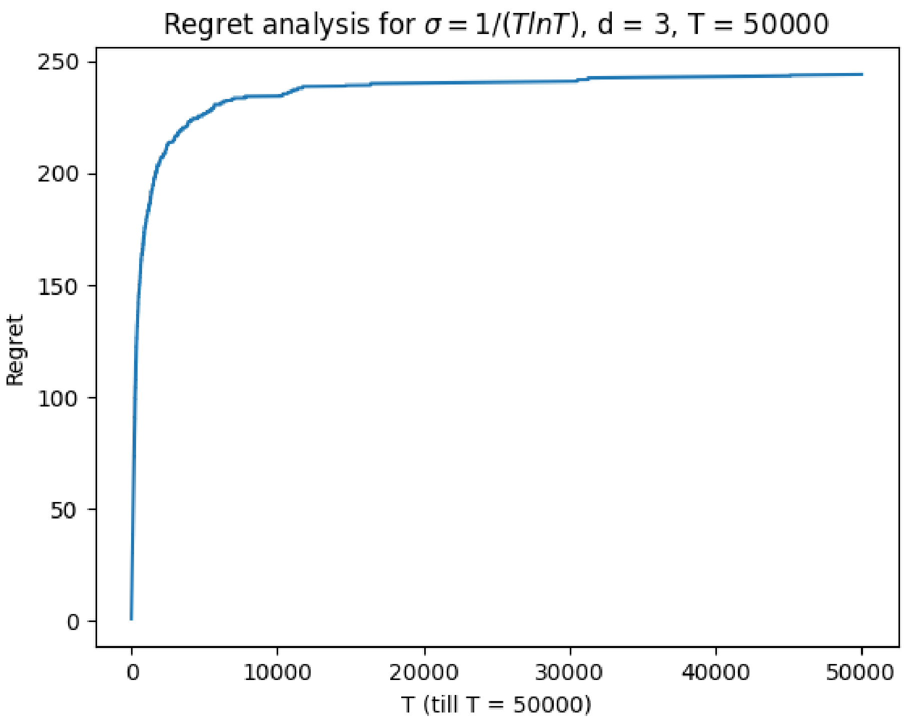
Figure 12 Regret of EllipsoidPricing and EXP4 as a function of $\ln T$ for different values of d. Note that the scales of the y-axes are different in the two plots.

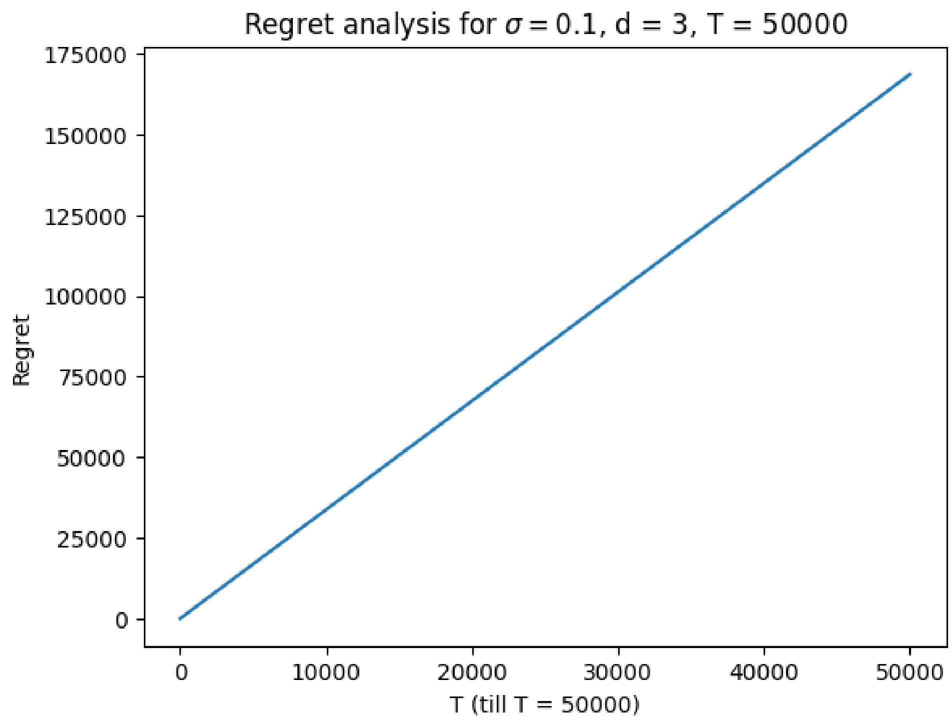
ELLIPSOID PRICING vs EXP4 Algorithm (our work):



Regret of EllipsoidPricing and EXP4 as a function of lnT for different values of d.

SHALLOW PRICING Regret analysis (our work):





FUTURE RESEARCH DIRECTION

Future Research Direction

- The best-known regret for one dimensional case was $O(lnlnT)$ as proposed by Klenberg and Leighton (2003). This implies that regret bound of $\Omega(dlnlnT)$ can be possible in d-dimensional case.
- EllipsoidEXP4 requires bound on the noise parameter σ to be known. Unknown σ with a Gaussian noise is tractable. However, σ-sub-Gaussian is yet to be resolved.
- Setting where parameter θ varies with time, but under the assumption of limited variation, is another possible area of research.
- Contextual Dynamic Pricing with Strategic Buyers (Liu, (2023)) proposes that the problem adversarial setting is an interesting direction of future research.

Thanks