Feature-Based Dynamic Pricing

Maxime Cohen $^{1,2}$  Ilan Lobel $^{1,2}$  Renato Paes Leme $^{2}$

$^{1}$NYU Stern

$^{2}$Google Research
Real estate agent problem

In each timestep the real estate agent receives a house to sell and needs to decide which price to put it in the market.

**Setup:** In each timestep:

1. Receives an item with feature vector $x_t \in \mathbb{R}^d$.
2. Chooses a price $p_t$ for the house.
3. Observes if the house was sold or not.
   - If $p_t \leq v(x_t)$, we sell and make profit $p_t$.
   - If $p_t > v(x_t)$, we don’t sell and make zero profit.
Real estate agent problem

In each timestep the real estate agent receives a house to sell and needs to decide which price to put it in the market.

**Setup:** In each timestep:

1. Receives an item with feature vector $x_t \in \mathbb{R}^d$.
   e.g. $x_t = (2$ bedroom, $1$ bathroom, no fireplace, Brooklyn, ...)$
Real estate agent problem

In each timestep the real estate agent receives a house to sell and needs to decide which price to put it in the market.

Setup: In each timestep:

1. Receives an item with feature vector $x_t \in \mathbb{R}^d$.
   e.g. $x_t = (2$ bedroom, $1$ bathroom, no fireplace, Brooklyn, ...)
Real estate agent problem

In each timestep the real estate agent receives a house to sell and needs to decide which price to put it in the market.

Setup: In each timestep:

1. Receives an item with feature vector $x_t \in \mathbb{R}^d$.
   e.g. $x_t = (2, 1, 0, 1, ..)$
Real estate agent problem

In each timestep the real estate agent receives a house to sell and needs to decide which price to put it in the market.

Setup: In each timestep:

1. Receives an item with feature vector $x_t \in \mathbb{R}^d$.
   e.g. $x_t = (2, 1, 0, 1, ..)$

2. Chooses a price $p_t$ for the house.

  ▶ if $p_t \leq v(x_t)$, we sell and make profit $p_t$.
  ▶ if $p_t > v(x_t)$, we don't sell and make zero profit.
Real estate agent problem

In each timestep the real estate agent receives a house to sell and needs to decide which price to put it in the market.

Setup: In each timestep:

1. Receives an item with feature vector $x_t \in \mathbb{R}^d$.
   e.g. $x_t = (2, 1, 0, 1, ..)$

2. Chooses a price $p_t$ for the house.

3. Observes if the house was sold or not.
Real estate agent problem

In each timestep the real estate agent receives a house to sell and needs to decide which price to put it in the market.

Setup: In each timestep:
1. Receives an item with feature vector $x_t \in \mathbb{R}^d$.
   e.g. $x_t = (2, 1, 0, 1, ..)$
2. Chooses a price $p_t$ for the house.
3. Observes if the house was sold or not.
   ▶ if $p_t \leq v(x_t)$, we sell and make profit $p_t$.
   ▶ if $p_t > v(x_t)$, we don't sell and make zero profit.
Challenges and Assumptions

Learn/Earn or Explore/Exploit:
We don’t know the market value $v(x_t)$.

Contextual problem:
The product is different in each round and adversarially chosen.

\[ v(x_t) = \theta^\top x_t \]
\[ \parallel x_t \parallel \leq 1, \quad \forall t, \parallel \theta \parallel \leq R. \]
Challenges and Assumptions

Learn/Earn or Explore/Exploit:
   We don’t know the market value $v(x_t)$.

Contextual problem:
   The product is different in each round and adversarially chosen.

Assumptions:
Challenges and Assumptions

Learn/Earn or Explore/Exploit:
We don’t know the market value \( v(x_t) \).

Contextual problem:
The product is different in each round and adversarially chosen.

Assumptions:
1. Linear model: \( v(x_t) = \theta^\top x_t \) for \( \theta \in \mathbb{R}^d \).
Challenges and Assumptions

Learn/Earn or Explore/Exploit:
We don’t know the market value $v(x_t)$.

Contextual problem:
The product is different in each round and adversarially chosen.

Assumptions:
1. Linear model: $v(x_t) = \theta^T x_t$ for $\theta \in \mathbb{R}^d$.
2. The parameter $\theta$ is unknown but fixed.
Challenges and Assumptions

Learn/Earn or Explore/Exploit:
We don’t know the market value \( v(x_t) \).

Contextual problem:
The product is different in each round and adversarially chosen.

Assumptions:
1. Linear model: \( v(x_t) = \theta^\top x_t \) for \( \theta \in \mathbb{R}^d \).
2. The parameter \( \theta \) is unknown but fixed.
3. Normalization: \( \|x_t\| \leq 1, \forall t, \|\theta\| \leq R \).
Goal and Applications

**Goal:** Minimize worst-case regret.

\[
\text{Regret} = \sum_{t=1}^{T} \theta^\top x_t - p_t \cdot 1\{p_t \leq \theta^\top x_t\}
\]

**Applications:** online advertisement, real-estate, domain pricing, ...
Non-contextual setting

**Simple setting:**
One dimensional \((d = 1)\) + no context \(x_t = 1, \forall t\).
Regret = \(\theta T - \sum_t p_t \cdot 1\{p_t \leq \theta\}\). and \(\theta \in [0, 1]\).
Non-contextual setting

**Simple setting:**
One dimensional \((d = 1)\) + no context \(x_t = 1, \forall t\).
Regret = \(\theta T - \sum_t p_t \cdot 1\{p_t \leq \theta\}\). and \(\theta \in [0, 1]\).

**Binary search:**

\[K_0 = \begin{array}{c}
0 \\
1
\end{array}\]
Non-contextual setting

**Simple setting:**
One dimensional \((d = 1)\) + no context \(x_t = 1, \forall t\).
Regret = \(\theta T - \sum_t p_t \cdot 1\{p_t \leq \theta\}\). and \(\theta \in [0, 1]\).

**Binary search:**

\[
K_0 = \begin{array}{c}
0 \\
p_1 \\
1
\end{array}
\]
Non-contextual setting

**Simple setting:**

One dimensional \((d = 1)\) + no context \(x_t = 1, \forall t\).

Regret = \(\theta T - \sum_t p_t \cdot 1\{p_t \leq \theta\}\). and \(\theta \in [0, 1]\).

**Binary search:**

\[K_1 = \begin{align*}
0 & \quad \quad p_1 & \quad 1 \\
\text{don’t sell}
\end{align*}\]
Non-contextual setting

**Simple setting:**
One dimensional \((d = 1) + \) no context \(x_t = 1, \forall t\).
Regret = \(\theta T - \sum_t p_t \cdot 1\{p_t \leq \theta\}\). and \(\theta \in [0, 1]\).

**Binary search:**

\[
K_1 = \begin{array}{cccc}
0 & p_2 & p_1 & 1
\end{array}
\]

▶ after \(\log(1/\epsilon)\) rounds we know \(\hat{\theta} \in [\hat{\theta}, \hat{\theta} + \epsilon]\).

▶ so \(\hat{\theta}\) always sells so:
\[
\text{Regret} \leq \log \frac{1}{\epsilon} + (T - \log \frac{1}{\epsilon}) \cdot \epsilon = O(\log T)\text{ for }\epsilon = O(\log T/T).
\]

Leighton & Kleinberg: Optimal Regret = \(O(\log \log T)\).
Non-contextual setting

**Simple setting:**
One dimensional ($d = 1$) + no context $x_t = 1, \forall t$.
Regret $= \theta T - \sum_t p_t \cdot 1\{p_t \leq \theta\}$ and $\theta \in [0, 1]$.

**Binary search:**

$$K_2 = \begin{array}{cccc}
0 & p_2 & p_1 & 1 \\
\end{array}$$

After $\log(1/\epsilon)$ rounds we know $\theta \in [\hat{\theta}, \hat{\theta} + \epsilon]$.

So $\hat{\theta}$ always sells so:
Regret $\leq \log 1/\epsilon + (T - \log 1/\epsilon) \cdot \epsilon = O(\log T)$
for $\epsilon = O(\log T / T)$.

Leighton & Kleinberg: Optimal Regret $= O(\log \log T)$.
Non-contextual setting

Simple setting:
One dimensional ($d = 1$) + no context $x_t = 1, \forall t$.
Regret $= \theta T - \sum_t p_t \cdot 1\{p_t \leq \theta\}$. and $\theta \in [0, 1]$.

Binary search:

\[
K_2 = \begin{array}{cccc}
0 & p_2 & p_1 & 1 \\
\end{array}
\]

- after log($1/\epsilon$) rounds we know $\theta \in [\hat{\theta}, \hat{\theta} + \epsilon]$.
- so $\hat{\theta}$ always sells so:

\[
\text{Regret} \leq \log \frac{1}{\epsilon} + \left( T - \log \frac{1}{\epsilon} \right) \cdot \epsilon = O(\log T)
\]

for $\epsilon = O(\log T / T)$. 

Leighton & Kleinberg: Optimal Regret $= O(\log \log T)$.
Non-contextual setting

Simple setting:
One dimensional ($d = 1$) + no context $x_t = 1, \forall t$. 
Regret = $\theta T - \sum_t p_t \cdot 1\{p_t \leq \theta\}$. and $\theta \in [0, 1]$.

Binary search:

$$K_2 = \begin{array}{ccc}
0 & p_2 & p_1 & 1 \\
\end{array}$$

- after $\log(1/\epsilon)$ rounds we know $\theta \in [\hat{\theta}, \hat{\theta} + \epsilon]$.
- so $\hat{\theta}$ always sells so:

$$\text{Regret} \leq \log \frac{1}{\epsilon} + \left( T - \log \frac{1}{\epsilon} \right) \cdot \epsilon = O(\log T)$$

for $\epsilon = O(\log T / T)$.

- Leighton & Kleinberg: Optimal Regret = $O(\log \log T)$. 
Contextual Setting : Knowledge Sets

Knowledge sets $K_t$

All $\theta$ compatible with observations so far.
Contextual Setting: Knowledge Sets

Knowledge sets $K_t$

All $\theta$ compatible with observations so far.
Knowledge sets $K_t$
All $\theta$ compatible with observations so far.
Contextual Setting : Knowledge Sets

**Knowledge sets** $K_t$

All $\theta$ compatible with observations so far.

\[ p_t = \min_{\theta \in K_t} \theta^\top x_t \text{ (exploit price, always sells)} \]
\[ p_t = \max_{\theta \in K_t} \theta^\top x_t \text{ (never sells)} \]

$K_t$:
- \[ \theta^\top x_t < p_t \]
- \[ \theta^\top x_t \geq p_t \]
**Contextual Setting: Knowledge Sets**

**Knowledge sets** $K_t$

All $\theta$ compatible with observations so far.

\[ p_t = \min_{\theta \in K_t} \theta^\top x_t \text{ (exploit price, always sells)} \]

\[ p_t = \max_{\theta \in K_t} \theta^\top x_t \text{ (never sells)} \]

\[ x_t \begin{cases} \geq p_t & \text{sell} \\ < p_t & \text{don't sell} \end{cases} \]
Contextual Setting: Knowledge Sets

Knowledge sets $K_t$
All $\theta$ compatible with observations so far.
Knowledge sets $K_t$

All $\theta$ compatible with observations so far.

Price ranges $p_t \in [p_t, \bar{p}_t]$

$p_t = \min_{\theta \in K_t} \theta^\top x_t$ (exploit price, always sells)

$\bar{p}_t = \max_{\theta \in K_t} \theta^\top x_t$ (never sells)
Game: multi-dimensional binary search

Our Goal: Find $\hat{\theta}$ such that $\|\theta - \hat{\theta}\| \leq \epsilon$, since $|\theta^\top x_t - \hat{\theta}^\top x_t| \leq \epsilon$ for all contexts $x_t$. 
Game: multi-dimensional binary search

Our Goal: Find $\hat{\theta}$ such that $\|\theta - \hat{\theta}\| \leq \epsilon$, since $|\theta^\top x_t - \hat{\theta}^\top x_t| \leq \epsilon$ for all contexts $x_t$. 


Game: multi-dimensional binary search

Our Goal: Find $\hat{\theta}$ such that $\|\theta - \hat{\theta}\| \leq \epsilon$, since $|\theta^\top x_t - \hat{\theta}^\top x_t| \leq \epsilon$ for all contexts $x_t$. 
Game: multi-dimensional binary search

Our Goal: Find $\hat{\theta}$ such that $\|\theta - \hat{\theta}\| \leq \epsilon$, since $|\theta^\top x_t - \hat{\theta}^\top x_t| \leq \epsilon$ for all contexts $x_t$. 
Our Goal: Find $\hat{\theta}$ such that $\|\theta - \hat{\theta}\| \leq \epsilon$, since $|\theta^\top x_t - \hat{\theta}^\top x_t| \leq \epsilon$ for all contexts $x_t$. 
Game: multi-dimensional binary search

Our Goal: Find \( \hat{\theta} \) such that \( \|\theta - \hat{\theta}\| \leq \epsilon \), since \( |\theta^\top x_t - \hat{\theta}^\top x_t| \leq \epsilon \) for all contexts \( x_t \).
Game: multi-dimensional binary search

Our Goal: Find $\hat{\theta}$ such that $\|\theta - \hat{\theta}\| \leq \epsilon$, since $|\theta^\top x_t - \hat{\theta}^\top x_t| \leq \epsilon$ for all contexts $x_t$. 
Game: multi-dimensional binary search

Our Goal: Find $\hat{\theta}$ such that $\|\theta - \hat{\theta}\| \leq \epsilon$, since $|\theta^\top x_t - \hat{\theta}^\top x_t| \leq \epsilon$ for all contexts $x_t$.
Our Goal: Find $\hat{\theta}$ such that $\|\theta - \hat{\theta}\| \leq \epsilon$, since $|\theta^\top x_t - \hat{\theta}^\top x_t| \leq \epsilon$ for all contexts $x_t$. 
Game: multi-dimensional binary search

Our Goal: Find $\hat{\theta}$ such that $\|\theta - \hat{\theta}\| \leq \epsilon$, since $|\theta^\top x_t - \hat{\theta}^\top x_t| \leq \epsilon$ for all contexts $x_t$. 
Idea # 1

Plan:
Narrow down $K_t$ to $B(\hat{\theta}, \epsilon)$ and exploit from then on.

Issues with this approach:
- You may never see a certain feature.
- Some features might be correlated.
- Often it is not good to wait to profit.
Idea # 2

Plan:
Explore if there is enough uncertainty about $\theta^T x_t$.
Compute $\bar{p}_t = \max_{\theta \in K_t} \theta^T x_t$ and $p_t = \min_{\theta \in K_t} \theta^T x_t$
and exploit if
\[ |\bar{p}_t - p_t| \leq \epsilon \]

Which price to use in exploration:
From 1-dimensional binary search, we can try:
\[ p_t = \frac{1}{2} (\bar{p}_t + p_t) \]
Idea # 2

Plan:
Explore if there is enough uncertainty about $\theta^T x_t$.
Compute $\overline{p}_t = \max_{\theta \in K_t} \theta^T x_t$ and $\underline{p}_t = \min_{\theta \in K_t} \theta^T x_t$
and exploit if

$$|\overline{p}_t - \underline{p}_t| \leq \varepsilon$$

Which price to use in exploration:
From 1-dimensional binary search, we can try:

$$p_t = \frac{1}{2} \left( \overline{p}_t + \underline{p}_t \right)$$

Thm: Regret of this approach is exponential in $d$.

Intuition: Shaving corners of a polytope in higher dimensions.
Idea # 3

Plan:  
Choose the price to split $K_t$ in two halves of equal volume.

Issues with this approach:

- Not easily computable.
Idea # 3

Plan:

Choose the price to split $K_t$ in two halves of equal volume.

Issues with this approach:

- Not easily computable.
- I don’t know if it works or not.
- We get $K_t$ of small volume: $\text{vol}(K_t) \leq 2^{-t}$.
  What we want is $K_t \subseteq B(\hat{\theta}, \epsilon)$
Solution: Ellipsoids

Solution:
After cutting $K_t$ regularize to its Löwner-John ellipsoid (same idea as in the Ellipsoid Method).
Solution: Ellipsoids

Solution:
After cutting $K_t$ regularize to its Löwner-John ellipsoid (same idea as in the Ellipsoid Method).
Solution: Ellipsoids

Solution:
After cutting $K_t$ regularize to its Löwner-John ellipsoid (same idea as in the Ellipsoid Method).
Solution: Ellipsoids

Solution:
After cutting $K_t$ regularize to its Löwner-John ellipsoid (same idea as in the Ellipsoid Method).
Solution: Ellipsoids

Solution:
After cutting $K_t$ regularize to its Löwner-John ellipsoid (same idea as in the Ellipsoid Method).

- We are keeping in the knowledge set some region that are known not to contain $\theta$.
- Ellipsoids are simpler to control. We have a better grasp of them since they can be described by a simple formula:

$$E = \left\{ \theta \in \mathbb{R}^d; (\theta - \theta_0)\top A^{-1}(\theta - \theta_0) \leq 1 \right\}$$

for a positive definite matrix $A$. 
Learning Algorithm

Initialize $A_0 = I / \sqrt{R}$ and $\theta_0 = 0$, i.e. $K_0 = B(0, R)$. Implicitly we keep $K_t = \{ \theta; (\theta - \theta_t)^\top A_t^{-1}(\theta - \theta_t) \leq 1 \}$

For each timestep $t$:

- Receive feature vector $x_t \in \mathbb{R}^d$.
- Compute $p_t = \min_{\theta \in K_t} \theta^\top x_t$ and $\bar{p}_t = \max_{\theta \in K_t} \theta^\top x_t$.
- If $\bar{p}_t - p_t < \epsilon$ pick price $p_t = p_t$ (Exploit)
- Otherwise choose $p_t = \frac{1}{2} \left( \bar{p}_t + p_t \right)$ (Explore) and update:

$$A_{t+1} = \frac{d^2}{d^2 + 1} \left( A_t - \frac{2}{d + 1} bb^\top \right)$$

and $\theta_{t+1} = \theta_t \pm \frac{1}{d+1} b$ where $b = -\theta_t + \arg\max_{\theta \in K_t} \theta^\top x_t$. 
Learning Algorithm

Initialize $A_0 = I / \sqrt{R}$ and $\theta_0 = 0$, i.e. $K_0 = B(0, R)$. Implicitly we keep $K_t = \{\theta; (\theta - \theta_t)^\top A_t^{-1}(\theta - \theta_t) \leq 1\}$

For each timestep $t$:

- Receive feature vector $x_t \in \mathbb{R}^d$.
- Compute $p_t = \min_{\theta \in K_t} \theta^\top x_t$ and $\bar{p}_t = \max_{\theta \in K_t} \theta^\top x_t$. (Solving a linear system since $K_t$ is an ellipsoid)
- If $\bar{p}_t - p_t < \epsilon$ pick price $p_t = p_t$ (Exploit)
- Otherwise choose $p_t = \frac{1}{2} \left( \bar{p}_t + p_t \right)$ (Explore) and update:

$$A_{t+1} = \frac{d^2}{d^2 + 1} \left( A_t - \frac{2}{d+1} bb^\top \right)$$

and $\theta_{t+1} = \theta_t \pm \frac{1}{d+1} b$ where $b = -\theta_t + \arg\max_{\theta \in K_t} \theta^\top x_t$.
**Main Theorem**

**Strategy for proving low regret**

Guarantee a small number of exploration steps.

**Lemma:** If we explore for more than \( O \left( Rd^2 \log \left( \frac{Rd}{e^2} \right) \right) \) steps, then \( K_t \) will be contained in a ball of radius \( \epsilon \). From then on, the algorithm will only exploit.

**Theorem:** Regret \( \leq O(Rd^2 \log T) \) for \( \epsilon = Rd^2 / T \).
Proof strategy

We know $\text{vol}(K_{t+1}) \leq e^{-1/(d+1)}\text{vol}(K_t)$. 
Proof strategy

- We know $\text{vol}(K_{t+1}) \leq e^{-1/(d+1)} \text{vol}(K_t)$.
- We need a bound on the width, which is $\max_{\theta \in K_t} \theta^T x - \min_{\theta \in K_t} \theta^T x$. Corresponds to bounding the eigenvalues of $A_t$. 
Proof strategy

- We know \( \text{vol}(K_{t+1}) \leq e^{-1/(d+1)\text{vol}(K_t)} \).

- We need a bound on the width, which is \( \max_{\theta \in K_t} \theta^\top x - \min_{\theta \in K_t} \theta^\top x \). Corresponds to bounding the eigenvalues of \( A_t \).

- We know \( \text{vol}_t = c_d \cdot \sqrt{\prod_i \lambda_i^t} = e^{-t/(d+1)} \). If we show that the smallest eigenvalue doesn’t decrease too fast, then all the eigenvalues must eventually be small.
Proof strategy

- We know $\text{vol}(K_{t+1}) \leq e^{-1/(d+1)}\text{vol}(K_t)$.
- We need a bound on the width, which is $\max_{\theta \in K_t} \theta^\top x - \min_{\theta \in K_t} \theta^\top x$. Corresponds to bounding the eigenvalues of $A_t$.
- We know $\text{vol}_t = c_d \cdot \sqrt{\prod_i \lambda_i^t} = e^{-t/(d+1)}$. If we show that the smallest eigenvalue doesn’t decrease too fast, then all the eigenvalues must eventually be small.
- We need to use the fact we never cut along directions that have small width, where width $= \overline{\rho}_t - \underline{p}_t$. 
Controlling eigenvalues (high level details)

- Given eigenvalue of $A_t$ we want to bound the eigenvalues of

$$A_{t+1} = \frac{d^2}{d^2 + 1} \left( A_t - \frac{2}{d + 1} bb^\top \right)$$

$$B_{t+1}$$

- If $\lambda_1^t \geq \ldots \geq \lambda_d^t$ are the eigenvalues of $A_t$, then the characteristic polynomial of $B_{t+1}$ is:

$$\varphi_{B_{t+1}}(x) = \prod_j (\lambda_j - x) \cdot \left[ 1 - \beta \sum_i \frac{\tilde{b}_i^2}{\lambda_i - x} \right]$$

$$\hat{\varphi}_{B_{t+1}}$$

- $\lambda_d^{t+1} \geq \lambda_d^t$ iff $\hat{\varphi}_{B_{t+1}} \left( \frac{d^2 - 1}{d^2} \lambda_d^t \right) \geq 0$. We show that this inequality holds whenever $\lambda_d^t$ is small enough and $b^\top x \geq \epsilon$. 
Connections

1. **Contextual Bandits**: We have a contextual bandit setting with adversarial context and a discontinuous loss function:

   \[
   \text{Regret} = \max_{\theta} \sum_{t=1}^{T} \theta^\top x_t \cdot 1\{\theta^\top x_t \leq v_t\} - p_t \cdot 1\{p_t \leq v_t\}
   \]

2. **Out of the shelf** contextual learning algorithms obtain \(O(\sqrt{T})\) regret, are more computationally expensive, but don’t assume that \(\theta\) is fixed, instead they seek to be competitive against the best \(\theta\):

3. **Quantum states** (?): Probing a buyer if he will buy at a certain price shares similarities with probing a quantum state with a linear measurement.
Lower bounds and Open Problems

1. A lower bound of $\Omega(d \log \log T)$ can be derived from embedding $d$ independent instances of the 1-dimensional problem (feature vectors are coordinate vectors).
2. Other applications of multi-dimensional binary search.
3. Stochastic setting: $\theta \sim \mathcal{F}$, $x \sim \mathcal{D}$. 
Thanks !