Corruption-Robust Contextual Search

Chara Podimata
(UC Berkeley $\rightarrow$ MIT )

Based on joint works with

D3P Semester, Fall22
Contextual Search Realizable Version

For rounds $t = 1, \ldots, T$: 

[Cohen, Lobel, Paes Leme, EC16/MS19], [Lobel, Paes Leme, Vladu, EC17/OR18], [Paes Leme, Schneider, FOCS18], [Liu, Paes Leme, Schneider, SODA21]
For rounds $t = 1, \ldots, T$:

1. Nature chooses $d$-dimensional context $u_t \in \mathbb{R}^d$, s.t., $||u_t|| = 1$. 

[Cohen, Lobel, Paes Leme, EC16/MS19], [Lobel, Paes Leme, Vladu, EC17/OR18], [Paes Leme, Schneider, FOCS18], [Liu, Paes Leme, Schneider, SODA21]
For rounds $t = 1, \ldots, T$:

1. Nature chooses $d$-dimensional context $u_t \in \mathbb{R}^d$, s.t., $\|u_t\| = 1$.
2. Learner queries a scalar $y_t \in \mathbb{R}$.

[Cohen, Lobel, Paes Leme, EC16/MS19], [Lobel, Paes Leme, Vladu, EC17/OR18], [Paes Leme, Schneider, FOCS18], [Liu, Paes Leme, Schneider, SODA21]
For rounds $t = 1, ..., T$:

1. Nature chooses $d$-dimensional context $u_t \in \mathbb{R}^d$, s.t., $||u_t|| = 1$.

2. Learner queries a scalar $y_t \in \mathbb{R}$.

3. Nature replies $\text{sign}(y_t - \langle u_t, \theta^* \rangle) \in \{-1, +1\}$ (binary feedback).
Contextual Search Realizable Version

“Realizable” = Exists hidden $\theta^* \in \mathbb{R}^d$ same across rounds.

For rounds $t = 1, ..., T$:

1. Nature chooses $d$-dimensional context $u_t \in \mathbb{R}^d$, s.t., $||u_t|| = 1$.
2. Learner queries a scalar $y_t \in \mathbb{R}$.
3. Nature replies $\text{sign}(y_t - \langle u_t, \theta^* \rangle) \in \{-1, +1\}$ (binary feedback).

[Cohen, Lobel, Paes Leme, EC16/MS19], [Lobel, Paes Leme, Vladu, EC17/OR18],
[Paes Leme, Schneider, FOCS18], [Liu, Paes Leme, Schneider, SODA21]
"Realizable" = Exists hidden $\theta^* \in \mathbb{R}^d$ same across rounds.

For rounds $t = 1, ..., T$:

1. Nature chooses $d$-dimensional context $u_t \in \mathbb{R}^d$, s.t., $||u_t|| = 1$.
2. Learner queries a scalar $y_t \in \mathbb{R}$.
3. Nature replies $\text{sign}(y_t - \langle u_t, \theta^* \rangle) \in \{-1, +1\}$ (binary feedback).
4. Learner incurs (but does not observe) loss $\ell(y_t, \langle u_t, \theta^* \rangle) \in [0,1]$.
Contextual Search Realizable Version

“Realizable” = Exists hidden $\theta^* \in \mathbb{R}^d$ same across rounds.

For rounds $t = 1, \ldots, T$:

1. Nature chooses $d$-dimensional context $u_t \in \mathbb{R}^d$, s.t., $||u_t|| = 1$.

2. Learner queries a scalar $y_t \in \mathbb{R}$.

3. Nature replies $\text{sign}(y_t - \langle u_t, \theta^*\rangle) \in \{-1, +1\}$ (binary feedback).

4. Learner incurs (but does not observe) loss $\ell(y_t, \langle u_t, \theta^*\rangle) \in [0,1]$.
   
   - $\epsilon$-ball loss: $\ell(y_t, y_t^*) = 1 \cdot 1\{|y_t - y_t^*| \geq \epsilon\}$
   - symmetric loss: $\ell(y_t, y_t^*) = |y_t - y_t^*|$  
   - pricing loss: $\ell(y_t, y_t^*) = y_t^* - y_t \cdot 1\{y_t \leq y_t^*\}$

[Cohen, Lobel, Paes Leme, EC16/MS19], [Lobel, Paes Leme, Vladu, EC17/OR18], [Paes Leme, Schneider, FOCS18], [Liu, Paes Leme, Schneider, SODA21]
Contextual Search Realizable Version

“Realizable” = Exists hidden $\theta^* \in \mathbb{R}^d$ same across rounds.

For rounds $t = 1, ..., T$:

1. Nature chooses $d$-dimensional context $u_t \in \mathbb{R}^d$, s.t., $||u_t|| = 1$.

2. Learner queries a scalar $y_t \in \mathbb{R}$.

3. Nature replies $\text{sign}(y_t - \langle u_t, \theta^* \rangle) \in \{-1, +1\}$ (binary feedback).

4. Learner incurs (but does not observe) loss $\ell(y_t, \langle u_t, \theta^* \rangle) \in [0,1]$.  

   - $\varepsilon$-ball loss: $\ell(y_t, y_t^*) = 1 \cdot 1\{y_t - y_t^* \geq \varepsilon\}$
   - Symmetric loss: $\ell(y_t, y_t^*) = |y_t - y_t^*|$
   - Pricing loss: $\ell(y_t, y_t^*) = y_t^* - y_t \cdot 1\{y_t \leq y_t^*\}$

$\text{Regret} = \text{total loss incurred} - \text{total loss for benchmark querying policy}$

[Cohen, Lobel, Paes Leme, EC16/MS19], [Lobel, Paes Leme, Vladu, EC17/OR18], [Paes Leme, Schneider, FOCS18], [Liu, Paes Leme, Schneider, SODA21]
Contextual Search Realizable Version

“Realizable” = Exists hidden $\theta^* \in \mathbb{R}^d$ same across rounds.
For rounds $t = 1, \ldots, T$:

1. Nature chooses $d$-dimensional context $u_t \in \mathbb{R}^d$, s.t., $||u_t|| = 1$.
2. Learner queries a scalar $y_t \in \mathbb{R}$.
3. Nature replies $\text{sign}(y_t - \langle u_t, \theta^* \rangle) \in \{-1, +1\}$ (binary feedback).
4. Learner incurs \textbf{(but does not observe)} loss $\ell(y_t, \langle u_t, \theta^* \rangle) \in [0,1]$.

- $\varepsilon$ – ball loss: $\ell(y_t, y_t^*) = 1 \cdot 1\{y_t - y_t^* \geq \varepsilon\}$
- symmetric loss: $\ell(y_t, y_t^*) = |y_t - y_t^*|$
- pricing loss: $\ell(y_t, y_t^*) = y_t^* - y_t \cdot 1\{y_t \leq y_t^*\}$

\textbf{Regret} = total loss incurred – total loss for benchmark querying policy

[Cohen, Lobel, Paes Leme, EC16/MS19], [Lobel, Paes Leme, Vladu, EC17/OR18], [Paes Leme, Schneider, FOCS18], [Liu, Paes Leme, Schneider, SODA21]
### Known Results

<table>
<thead>
<tr>
<th>Loss</th>
<th>$\ell(y_t, y_t^*)$</th>
<th>Lower Bound [R]</th>
<th>Upper Bound [R]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[Cohen, Lobel, Paes Leme, EC16/MS19], [Lobel, Paes Leme, Vladu, EC17/OR18], [Paes Leme, Schneider, FOCS18], [Liu, Paes Leme, Schneider, SODA21]
## Known Results

<table>
<thead>
<tr>
<th>Loss</th>
<th>$\ell(y_t, y_t^*)$</th>
<th>Lower Bound [R]</th>
<th>Upper Bound [R]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon$ -ball</td>
<td>$1{</td>
<td>y_t - y_t^*</td>
<td>\geq \varepsilon}$</td>
</tr>
</tbody>
</table>

[Cohen, Lobel, Paes Leme, EC16/MS19], [Lobel, Paes Leme, Vladu, EC17/OR18], [Paes Leme, Schneider, FOCS18], [Liu, Paes Leme, Schneider, SODA21]
## Known Results

<table>
<thead>
<tr>
<th>Loss</th>
<th>$\ell(y_t, y_t^*)$</th>
<th>Lower Bound [R]</th>
<th>Upper Bound [R]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon$-ball</td>
<td>$1 {</td>
<td>y_t - y_t^*</td>
<td>\geq \varepsilon}$</td>
</tr>
<tr>
<td>symmetric</td>
<td>$</td>
<td>y_t - y_t^*</td>
<td>$</td>
</tr>
</tbody>
</table>

[Cohen, Lobel, Paes Leme, EC16/MS19], [Lobel, Paes Leme, Vladu, EC17/OR18], [Paes Leme, Schneider, FOCS18], [Liu, Paes Leme, Schneider, SODA21]
### Known Results

<table>
<thead>
<tr>
<th>Loss</th>
<th>Loss Function</th>
<th>Lower Bound [R]</th>
<th>Upper Bound [R]</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \varepsilon )-ball</td>
<td>( 1{</td>
<td>y_t - y_t^*</td>
<td>\geq \varepsilon} )</td>
</tr>
<tr>
<td>symmetric</td>
<td>(</td>
<td>y_t - y_t^*</td>
<td>)</td>
</tr>
<tr>
<td>pricing</td>
<td>( y_t^* - y_t \cdot 1{y_t \leq y_t^*} )</td>
<td>( \Omega(d \log \log T) )</td>
<td>( O(d \log \log T + d \log d) ) [LLS21]</td>
</tr>
</tbody>
</table>

[Cohen, Lobel, Paes Leme, EC16/MS19], [Lobel, Paes Leme, Vladu, EC17/OR18], [Paes Leme, Schneider, FOCS18], [Liu, Paes Leme, Schneider, SODA21]
Contextual Search with Adversarial Noise

Exists hidden $\theta^* \in \mathbb{R}^d$ same across rounds.

For rounds $t = 1, \ldots, T$:

1. Nature chooses $d$-dimensional context $u_t \in \mathbb{R}^d$, s.t., $|u_t| = 1$.

2. Learner queries a scalar $y_t \in \mathbb{R}$.

3. Nature replies $\sigma_t = \text{sign}(y_t - y_t^*) \in \{-1, +1\}$ (binary feedback).

4. Learner incurs (but does not observe) loss $\ell(y_t, y_t^*) \in [0,1]$. 
Contextual Search with Adversarial Noise

Exists hidden $\theta^* \in \mathbb{R}^d$ same across rounds.

For rounds $t = 1, \ldots, T$:

1. Nature chooses $d$-dimensional context $u_t \in \mathbb{R}^d$, s.t., $||u_t|| = 1$.

2. Learner queries a scalar $y_t \in \mathbb{R}$.

3. Nature replies $\sigma_t = \text{sign}(y_t - y_t^*) \in \{-1, +1\}$ (binary feedback).

4. Learner incurs (but does not observe) loss $\ell(y_t, y_t^*) \in [0,1]$. 

\[ y_t^* = \langle u_t, \theta^* \rangle + z_t \]

\[ z_t \in [0,1]: \text{adaptively and adversarially chosen} \]
Contextual Search with Adversarial Noise

Exists hidden $\theta^* \in \mathbb{R}^d$ same across rounds.

For rounds $t = 1, \ldots, T$:

1. Nature chooses $d$-dimensional context $u_t \in \mathbb{R}^d$, s.t., $||u_t|| = 1$.

2. Learner queries a scalar $y_t \in \mathbb{R}$.

3. Nature replies $\sigma_t = \text{sign}(y_t - y_t^*) \in \{-1, +1\}$ (binary feedback).

4. Learner incurs (but does not observe) loss $\ell(y_t, y_t^*) \in [0, 1]$. 

$y_t^* = \langle u_t, \theta^* \rangle + z_t$ 
$z_t \in [-1, 1]$: adaptively and adversarially chosen
Contextual Search with Adversarial Noise

Exists hidden $\theta^* \in \mathbb{R}^d$ same across rounds.

For rounds $t = 1, \ldots, T$:

1. Nature chooses $d$-dimensional context $u_t \in \mathbb{R}^d$, s.t., $||u_t|| = 1$.

2. Learner queries a scalar $y_t \in \mathbb{R}$.

3. Nature replies $\sigma_t = \text{sign}(y_t - y_t^*) \in \{-1, +1\}$ (binary feedback).

4. Learner incurs (but does not observe) loss $\ell(y_t, y_t^*) \in [0,1]$. 

\[
y_t^* = \langle u_t, \theta^* \rangle + z_t
\]

$z_t \in [-1,1]$: adaptively and adversarially chosen
Contextual Search with Adversarial Noise

Exists hidden $\theta^* \in \mathbb{R}^d$ same across rounds.

For rounds $t = 1, \ldots, T$:

1. Nature chooses $d$-dimensional context $u_t \in \mathbb{R}^d$, s.t., $||u_t|| = 1$.
2. Learner queries a scalar $y_t \in \mathbb{R}$.
3. Nature replies $\sigma_t = \text{sign}(y_t - y_t^*) \in \{-1, +1\}$ (binary feedback).
4. Learner incurs (but does not observe) loss $\ell(y_t, y_t^*) \in [0,1]$.

Desiderata for our Algorithms
- ✓ Graceful degradation of regret with $C$
- ✓ No knowledge of $C$ assumed (agnostic)
Main Results

[Krishnamurthy, Lykouris, P., Schapire, STOC21/OR22]

$\varepsilon$ – ball loss: $\text{Regret} = O(C_0 \, d^3 \, \log^3 \, 1/\varepsilon)$

symmetric, pricing loss: $\text{Regret} = O(C_0 \, d^3 \, \log^3 \, T)$

[Paes Leme, P., Schneider, COLT22]

$\varepsilon$ – ball loss: $\text{Regret} = O(C_0 + d \, \log \, 1/\varepsilon)$

symmetric loss: polytime, $\text{Regret} = O(C_1 + d \, \log \, T)$, where $C_1 = \Sigma_t |z_t|$
## Known Results

<table>
<thead>
<tr>
<th>Loss</th>
<th>$\ell(y_t, y_t^*)$</th>
<th>Lower Bound [R]</th>
<th>Upper Bound [R]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon$-ball</td>
<td>$1{</td>
<td>y_t - y_t^*</td>
<td>\geq \varepsilon}$</td>
</tr>
<tr>
<td>symmetric</td>
<td>$</td>
<td>y_t - y_t^*</td>
<td>$</td>
</tr>
<tr>
<td>pricing</td>
<td>$y_t^* - y_t \cdot 1{y_t \leq y_t^*}$</td>
<td>$\Omega(d \log \log T)$</td>
<td>$O(d \log \log T + d \log d)$ [LLS21]</td>
</tr>
</tbody>
</table>

[Cohen, Lobel, Paes Leme, EC16/MS19], [Lobel, Paes Leme, Vladu, EC17/OR18], [Paes Leme, Schneider, FOCS18], [Liu, Paes Leme, Schneider, SODA21]
Contextual Search with Adversarial Noise

Exists hidden $\theta^* \in \mathbb{R}^d$ same across rounds.

For rounds $t = 1, \ldots, T$:

1. Nature chooses $d$-dimensional context $u_t \in \mathbb{R}^d$, s.t., $||u_t|| = 1$.

2. Learner queries a scalar $y_t \in \mathbb{R}$.

3. Nature replies $\sigma_t = \text{sign}(y_t - y_t^*) \in \{-1, +1\}$ (binary feedback).

4. Learner incurs (but does not observe) loss $\ell(y_t, y_t^*) \in [0,1]$.

for $z_t \in \{0,1\}$:

- $\text{Regret} = O(C_0 \ d^3 \ \log^3 T)$ for pricing, symmetric loss
- $\text{Regret} = O(C_0 \ d^3 \ \log^3 1/\epsilon)$ for $\epsilon$-ball loss

[Krishnamurthy, Lykouris, P., Schapire, STOC21/OR22]

Runtime $\text{poly} (d, \log T)^{\text{poly}(\log T)}$
Traditional Approach for Contextual Search Algorithms

Traditional Approach for Contextual Search Algorithms

For rounds $t = 1, \ldots, T$:
Traditional Approach for Contextual Search Algorithms

For rounds $t = 1, \ldots, T$:

- Maintain active **knowledge set** with feasible values for $\theta^*$.
Traditional Approach for Contextual Search Algorithms

For rounds $t = 1, \ldots, T$:

- Maintain active **knowledge set** with feasible values for $\theta^*$.  

- Learner chooses $y_t$ to make enough "progress"
  
  (e.g., $y_t = \langle u_t, \text{centroid of knowledge set} \rangle$).

Knowledge set for $\theta^*$
Traditional Approach for Contextual Search Algorithms

For rounds $t = 1, \ldots, T$:

- Maintain active **knowledge set** with feasible values for $\theta^*$.
- Learner chooses $y_t$ to make enough "progress"
  (e.g., $y_t = \langle u_t, \text{centroid of knowledge set} \rangle$).
- Eliminate **inconsistent side** of knowledge set.
Traditional Approach for Contextual Search Algorithms

For rounds $t = 1, \ldots, T$:

- Maintain active **knowledge set** with feasible values for $\theta^*$.
- Learner chooses $y_t$ to make enough "progress" (e.g., $y_t = \langle u_t, \text{centroid of knowledge set} \rangle$).
- Eliminate **inconsistent side** of knowledge set.

[CoHEN, LOBEL, PAES LEME, EC16/MS19]. [LOBEL, PAES LEME, VLADU, EC17/OR18], [PAES LEME, SCHNEIDER, FOCS18], [LIU, PAES LEME, SCHNEIDER, SODA21]
Traditional Approach for Contextual Search Algorithms

For rounds $t = 1, \ldots, T$:

- Maintain active **knowledge set** with feasible values for $\theta^*$.  
- Learner chooses $y_t$ to make enough "**progress**"  
  (e.g., $y_t = \langle u_t, \text{centroid of knowledge set} \rangle$).  
- Eliminate **inconsistent side** of knowledge set.

[Evgeny Cohen, Ilan Lobel, Rafael Paes Leme, EC16/MS19],  
[Ilan Lobel, Rafael Paes Leme, Vladu, EC17/OR18],  
[Rafael Paes Leme, Yishay Schneider, FOCS18],  
[Zhen Liu, Rafael Paes Leme, Yishay Schneider, SODA21]
Traditional Approach for Contextual Search Algorithms

For rounds $t = 1, \ldots, T$:

- Maintain active **knowledge set** with feasible values for $\theta^*$. 
- Learner chooses $y_t$ to make enough "progress" (e.g., $y_t = \langle u_t, \text{centroid of knowledge set} \rangle$).
- Eliminate **inconsistent side** of knowledge set.

[Cohe, Lobel, Paes Leme, EC16/MS19], [Lobel, Paes Leme, Vladu, EC17/OR18],
[Paes Leme, Schneider, FOCS18], [Liu, Paes Leme, Schneider, SODA21]
Traditional Approach for Contextual Search Algorithms

For rounds $t = 1, \ldots, T$:

- Maintain active **knowledge set** with feasible values for $\theta^*$.
- Learner chooses $y_t$ to make enough "progress" (e.g., $y_t = \langle u_t, \text{centroid of knowledge set} \rangle$).
- Eliminate **inconsistent side** of knowledge set.

Knowledge set for $\theta^*$

[Aggressively introducing cuts $\rightarrow$ fast, logarithmic bounds]

[Cohen, Lobel, Paes Leme, EC16/MS19], [Lobel, Paes Leme, Vladu, EC17/OR18], [Paes Leme, Schneider, FOCS18], [Liu, Paes Leme, Schneider, SODA21]
Traditional Approach for Contextual Search Algorithms

For rounds $t = 1, \ldots, T$:

- Maintain active **knowledge set** with feasible values for $\theta^*$.
- Learner chooses $y_t$ to make enough "progress"
  (e.g., $y_t = \langle u_t, \text{centroid of knowledge set} \rangle$).
- Eliminate **inconsistent side** of knowledge set.

\[ \text{Regret} \leq O(d \cdot \log T) \]
Traditional Approach for Contextual Search Algorithms

For rounds $t = 1, \ldots, T$:

- Maintain active **knowledge set** with feasible values for $\theta^*$.  
- Learner chooses $y_t$ to make enough "progress"  
  (e.g., $y_t = \langle u_t, \text{centroid of knowledge set} \rangle$).  
- Eliminate **inconsistent side** of knowledge set.

**Important properties of cut**

1. **Never eliminate** $\theta^*$  
   - Retain all parameters consistent with feedback
2. **Volumetric progress**  
   - Cut through centroid

**Aggressively introducing cuts** $\Rightarrow$ fast, logarithmic bounds

\[ \text{Regret} \leq O(d \cdot \log T) \]
Idea Overview

1. Create a robust version of the knowledge-set-based algorithm that is robust to a known amount of corruption $\tilde{c} \approx \log T$.

2. **Unknown $C$**: Run a variant of the Multi-Layering Race technique from [Lykouris, Mirrokni, Paes Leme, *STOC18*].
Idea Overview

1. Create a robust version of the knowledge-set-based algorithm that is robust to a \textbf{known} amount of corruption $\tilde{c} \approx \log T$.

2. \textbf{Unknown} $C$: Run a variant of the Multi-Layering Race technique from [Lykouris, Mirrokni, Paes Leme, STOC18].
Robust Volumetric Progress
Robust Volumetric Progress

**Challenge 1**

We cannot repeat the same query (contexts are different at different rounds).
Robust Volumetric Progress

Challenge 1

We cannot repeat the same query (contexts are different at different rounds).

Context cut: hyperplane perpendicular to context

\[ u_1 = (1,0) \]
\[ u_2 = (0.8,0.6) \]
Robust Volumetric Progress

Challenge 1

We cannot repeat the same query (contexts are different at different rounds).

Context cut: hyperplane perpendicular to context

\[ u_1 = (1,0) \quad u_2 = (0.8, 0.6) \]

Idea 1

Keep “penalty” for each parameter & make cut once a context cut fully retains protected region on one side.

- Protected region: all parameters with “penalty” \( \leq \overline{c} \).
Robust Volumetric Progress

Challenge 1

We cannot repeat the same query (contexts are different at different rounds).

Context cut: hyperplane perpendicular to context

Idea 1

Keep “penalty” for each parameter & make cut once a context cut fully retains protected region on one side.

- Protected region: all parameters with “penalty” ≤ \( \tilde{c} \).

\[ u_1 = (1,0) \]
Robust Volumetric Progress

Challenge 1

We cannot repeat the same query (contexts are different at different rounds).

Context cut: hyperplane perpendicular to context

Idea 1

Keep “penalty” for each parameter & make cut once a context cut fully retains protected region on one side.

- Protected region: all parameters with “penalty” $\leq c$.

$u_1 = (1,0)$
Robust Volumetric Progress

**Challenge 1**
We cannot repeat the same query (contexts are different at different rounds).

**Context cut:** hyperplane perpendicular to context

- **Idea 1**
  - Keep “penalty” for each parameter & make cut once a context cut fully retains protected region on one side.
  - Protected region: all parameters with “penalty” ≤ $\tilde{c}$.

$u_1 = (1,0)$
Robust Volumetric Progress

Challenge 1
We cannot repeat the same query (contexts are different at different rounds).

Context cut: hyperplane perpendicular to context

Idea 1
Keep “penalty” for each parameter & make cut once a context cut fully retains protected region on one side.
• Protected region: all parameters with “penalty” ≤ $\bar{c}$.

$u_1 = (1,0)$
Robust Volumetric Progress

We cannot repeat the same query (contexts are different at different rounds).

**Challenge 1**

Context cut: hyperplane perpendicular to context

Keep “penalty” for each parameter & make cut once a context cut fully retains protected region on one side.

- Protected region: all parameters with “penalty” ≤ \( \bar{c} \).

\[ u_1 = (1,0) \]
Robust Volumetric Progress
Robust Volumetric Progress

Challenge 2

We may never have a context cut with the protected region fully on one side.
Robust Volumetric Progress

**Challenge 2**

We may never have a context cut with the protected region fully on one side.

**Counterexample:** Even with infinite contexts and $\bar{c} = 1$, no such context cut.
Robust Volumetric Progress

Challenge 2
We may never have a context cut with the protected region fully on one side.

Idea 2
Combine context cuts to compute a “valid cut”.

Counterexample: Even with infinite contexts and $\bar{c} = 1$, no such context cut.
Robust Volumetric Progress

Challenge 2
We may never have a context cut with the protected region fully on one side.

Counterexample: Even with infinite contexts and $\bar{c} = 1$, no such context cut.

Idea 2
Combine context cuts to compute a “valid cut”.

Important Properties of Valid Cut
1. Never eliminate $\theta^*$
   • Retains protected region on one side.
2. Volumetric progress
   • Cross close to centroid.
Robust Volumetric Progress

**Challenge 2**
We may never have a context cut with the protected region fully on one side.

**Counterexample**: Even with infinite contexts and \( \bar{c} = 1 \), no such context cut.

**Idea 2**
Combine context cuts to compute a “valid cut”.

**Idea 3**
Show that \( 2d \cdot (d + 1) \cdot \bar{c} + 1 \) context cuts have enough information to compute such a valid cut (Caratheodory’s theorem).

**Important Properties of Valid Cut**
1. Never eliminate \( \theta^* \)
   - Retains protected region on one side.
2. Volumetric progress
   - Cross close to centroid.
Robust Volumetric Progress

Challenge 2
We may never have a context cut with the protected region fully on one side.

Idea 2
Combine context cuts to compute a “valid cut”.

Idea 3
Show that $2d \cdot (d + 1) \cdot \bar{c} + 1$ context cuts have enough information to compute such a valid cut (Caratheodory’s theorem).

Counterexample: Even with infinite contexts and $\bar{c} = 1$, no such context cut.

Important Properties of Valid Cut
1. Never eliminate $\theta^*$
   - Retains protected region on one side.
2. Volumetric progress
   - Cross close to centroid.
Robust Volumetric Progress

**Challenge 2**
We may never have a context cut with the protected region fully on one side.

**Idea 2**
Combine context cuts to compute a “valid cut”.

**Idea 3**
Show that $2d \cdot (d + 1) \cdot \bar{c} + 1$ context cuts have enough information to compute such a valid cut (Caratheodory’s theorem).

**Counterexample:** Even with infinite contexts and $\bar{c} = 1$, no such context cut.

**Important Properties of Valid Cut**
1. Never eliminate $\theta^*$
   - Retains protected region on one side.
2. Volumetric progress
   - Cross close to centroid.

\[ \text{penalty} \leq \bar{c} \]
Robust Volumetric Progress

**Challenge 2**

We may never have a context cut with the protected region fully on one side.

**Idea 2**

Combine context cuts to compute a “valid cut”.

**Idea 3**

Show that \(2d \cdot (d + 1) \cdot \bar{c} + 1\) context cuts have enough information to compute such a valid cut (Caratheodory’s theorem).

**Counterexample:** Even with infinite contexts and \(\bar{c} = 1\), no such context cut.

**Important Properties of Valid Cut**

1. Never eliminate \(\theta^*\)
   - Retains protected region on one side.

2. Volumetric progress
   - Cross close to centroid.

\[\text{penalty} \leq \bar{c}\]
Robust Volumetric Progress

**Challenge 2**

We may never have a context cut with the protected region fully on one side.

**Idea 2**

Combine context cuts to compute a "valid cut".

**Idea 3**

Show that $2d \cdot (d + 1) \cdot \bar{c} + 1$ context cuts have enough information to compute such a valid cut (*Caratheodory’s theorem*).

**Counterexample:** Even with infinite contexts and $\bar{c} = 1$, no such context cut.

**Important Properties of Valid Cut**

1. *Never eliminate $\theta^*$*
   - Retains protected region on one side.
2. *Volumetric progress*
   - Cross close to centroid.

\[ \text{penalty} \leq \bar{c} \]
Robust Volumetric Progress

**Challenge 2**

We may never have a context cut with the protected region fully on one side.

**Counterexample:** Even with infinite contexts and $\bar{c} = 1$, no such context cut.

**Idea 2**

Combine context cuts to compute a “valid cut”.

**Idea 3**

Show that $2d \cdot (d + 1) \cdot \bar{c} + 1$ context cuts have enough information to compute such a valid cut (**Caratheodory’s theorem**).

**Important Properties of Valid Cut**

1. Never eliminate $\theta^*$
   - Retains protected region on one side.

2. Volumetric progress
   - Cross close to centroid.

**Known $\bar{c}$**

- **penalty $\leq \bar{c}$**
- $d + 1$ points
Robust Volumetric Progress

**Challenge 2**
We may never have a context cut with the protected region fully on one side.

**Counterexample:** Even with infinite contexts and $\bar{c} = 1$, no such context cut.

**Idea 2**
Combine context cuts to compute a “valid cut”.

**Idea 3**
Show that $2d \cdot (d + 1) \cdot \bar{c} + 1$ context cuts have enough information to compute such a valid cut (Caratheodory’s theorem).

**Important Properties of Valid Cut**
1. Never eliminate $\theta^*$
   - Retains protected region on one side.
2. Volumetric progress
   - Cross close to centroid.

Each penalty for black point attributed to $\geq 1$ protected point
$\Rightarrow$ penalty (black) $\leq \bar{c} \cdot (d + 1)$
Robust Volumetric Progress

**Challenge 2**
We may never have a context cut with the protected region fully on one side.

**Idea 2**
Combine context cuts to compute a "valid cut".

**Idea 3**
Show that $2d \cdot (d + 1) \cdot c + 1$ context cuts have enough information to compute such a valid cut (Caratheodory's theorem).

**Counterexample:** Even with infinite contexts and $\bar{c} = 1$, no such context cut.

**Important Properties of Valid Cut**

1. Never eliminate $\theta^*$
   - Retains protected region on one side.
2. Volumetric progress
   - Cross close to centroid.

Each penalty for black point attributed to $\geq 1$ protected point

$\text{penalty } \leq \bar{c}$

$\geq \bar{c} \cdot (d + 1) + 1$

$d + 1$ points

Known $\bar{c}$
Robust Volumetric Progress

**Challenge 2**

We may never have a context cut with the protected region fully on one side.

**Counterexample:** Even with infinite contexts and $\bar{c} = 1$, no such context cut.

**Idea 2**

Combine context cuts to compute a “valid cut”.

**Idea 3**

Show that $2d \cdot (d + 1) \cdot \bar{c} + 1$ context cuts have enough information to compute such a valid cut (*Caratheodory’s theorem*).

**Idea 4**

Use Perceptron to find a valid cut.

**Important Properties of Valid Cut**

1. Never eliminate $\theta^*$
   - Retains protected region on one side.
2. Volumetric progress
   - Cross close to centroid.
A Fundamentally Different Approach
A Fundamentally Different Approach

• Maintain \textit{probability density function} $f(\cdot)$ over all possible values of $\theta^*$.

• \textbf{Density} at point $x = \text{extent}$ to which $x$ is \textit{consistent} with $\theta^*$. 
A Fundamentally Different Approach

• Maintain **probability density function** $f(\cdot)$ over all possible values of $\theta^*$.  

• **Density** at point $x = \text{extent}$ to which $x$ is **consistent** with $\theta^*$.  

→ Never remove values from consideration, just shift its “weight”.  

→ Higher weight to more probable values.
A Fundamentally Different Approach

- Maintain **probability density function** $f(\cdot)$ over all possible values of $\theta^*$.
- **Density** at point $x = \text{extent}$ to which $x$ is **consistent** with $\theta^*$.

→ Never remove values from consideration, just shift its “weight”.
→ Higher weight to more probable values.
A Fundamentally Different Approach

• Maintain **probability density function** $f(\cdot)$ over all possible values of $\theta^*$.  
• **Density** at point $x = \text{extent}$ to which $x$ is **consistent** with $\theta^*$.  

→ Never remove values from consideration, just shift its “weight”.  
→ Higher weight to more probable values.
A Fundamentally Different Approach

• Maintain **probability density function** $f(\cdot)$ over all possible values of $\theta^*$.

• **Density** at point $x = \text{extent}$ to which $x$ is **consistent** with $\theta^*$.

→ Never remove values from consideration, just shift its “weight”.

→ Higher weight to more probable values.
A Fundamentally Different Approach

- Maintain **probability density function** \( f(\cdot) \) over all possible values of \( \theta^* \).

- **Density** at point \( x = \text{extent} \) to which \( x \) is **consistent** with \( \theta^* \).

→ Never remove values from consideration, just shift its “weight”.

→ Higher weight to more probable values.

Seemingly more ”forgiving” approach → faster bounds for corruption-robust
Algorithm for $\varepsilon$ – Ball Loss
Algorithm for $\varepsilon$ – Ball Loss

$\varepsilon$ – Window Median Algorithm

Initialize $f_1(x)$: uniform over $B(0,1)$. 
Algorithm for $\varepsilon$ – Ball Loss

$\varepsilon$ – Window Median Algorithm

Initialize $f_1(x)$: uniform over $B(0,1)$.

For rounds $t = 1, \ldots, T$: 

\[
\langle u_t, x \rangle = a
\]

$B(0,1)$
Algorithm for $\varepsilon$ – Ball Loss

**$\varepsilon$ – Window Median Algorithm**

Initialize $f_1(x)$: uniform over $B(0,1)$.

For rounds $t = 1, \ldots, T$:

- Observe $u_t$ and query $y_t = \varepsilon - \text{window - median}(f_t)$
Algorithm for $\varepsilon$ – Ball Loss

$\varepsilon$ – Window Median Algorithm

Initialize $f_1(x)$: uniform over $B(0,1)$.

For rounds $t = 1, \ldots, T$:

• Observe $u_t$ and query $y_t = \varepsilon - \text{window - median}(f_t)$
Algorithm for $\varepsilon$ – Ball Loss

$\varepsilon$ – Window Median Algorithm

Initialize $f_1(x)$: uniform over $B(0,1)$.

For rounds $t = 1, \ldots, T$:

- Observe $u_t$ and query $y_t = \varepsilon – \text{window – median}(f_t)$
Algorithm for $\varepsilon$ – Ball Loss

$\varepsilon$ – Window Median Algorithm

Initialize $f_1(x)$: uniform over $B(0,1)$.

For rounds $t = 1, \ldots, T$:

• Observe $u_t$ and query $y_t = \varepsilon - \text{window - median}(f_t)$
Algorithm for $\varepsilon$ – Ball Loss

$\varepsilon$ – Window Median Algorithm

Initialize $f_1(x)$: uniform over $B(0,1)$.

For rounds $t = 1, \ldots, T$:

- Observe $u_t$ and query $y_t = \varepsilon$ – window – median($f_t$)
Algorithm for \( \varepsilon \) – Ball Loss

\( \varepsilon \) – Window Median Algorithm

Initialize \( f_1(x) \): uniform over \( B(0,1) \).

For rounds \( t = 1, ..., T \):
- Observe \( u_t \) and query \( y_t = \varepsilon – \text{window – median}(f_t) \)
Algorithm for $\varepsilon$ – Ball Loss

$\varepsilon$ – Window Median Algorithm

Initialize $f_1(x)$: uniform over $B(0,1)$.

For rounds $t = 1, \ldots, T$:
- Observe $u_t$ and query $y_t = \varepsilon – \text{window – median}(f_t)$
- Update density:
\[
f_{t+1}(x) = \begin{cases} 
  \frac{3}{2} \cdot f_t(x), & \text{if } \sigma_t \cdot (\langle u_t, x \rangle - y_t) \geq \frac{\varepsilon}{2} \\
  1 \cdot f_t(x), & \text{if } -\frac{\varepsilon}{2} \leq \sigma_t \cdot (\langle u_t, x \rangle - y_t) \leq \frac{\varepsilon}{2} \\
  \frac{1}{2} \cdot f_t(x), & \text{if } \sigma_t \cdot (\langle u_t, x \rangle - y_t) \leq -\frac{\varepsilon}{2} \\
  \frac{1}{2} \cdot f_t(x), & \text{if } \sigma_t \cdot (\langle u_t, x \rangle - y_t) \leq \frac{\varepsilon}{2} 
\end{cases}
\]
Algorithm for $\varepsilon$ – Ball Loss

$\varepsilon$ – Window Median Algorithm

Initialize $f_1(x)$: uniform over $B(0,1)$.

For rounds $t = 1, \ldots, T$:

- Observe $u_t$ and query $y_t = \varepsilon – \text{window} – \text{median}(f_t)$
- Update density:

$$f_{t+1}(x) = \begin{cases} 
\frac{3}{2} \cdot f_t(x), & \text{if } \sigma_t \cdot (\langle u_t, x \rangle - y_t) \geq \frac{\varepsilon}{2} \\
1 \cdot f_t(x), & \text{if } -\frac{\varepsilon}{2} \leq \sigma_t \cdot (\langle u_t, x \rangle - y_t) \leq \frac{\varepsilon}{2} \\
\frac{1}{2} \cdot f_t(x), & \text{if } \sigma_t \cdot (\langle u_t, x \rangle - y_t) \leq -\frac{\varepsilon}{2}
\end{cases}$$
Algorithm for $\varepsilon$ – Ball Loss

$\varepsilon$ – Window Median Algorithm

Initialize $f_1(x)$: uniform over $B(0,1)$.

For rounds $t = 1, \ldots, T$:

- Observe $u_t$ and query $y_t = \varepsilon – window – median(f_t)$

- Update density:

$$f_{t+1}(x) = \begin{cases} 
\frac{3}{2} \cdot f_t(x), & \text{if } \sigma_t \cdot (\langle u_t, x \rangle - y_t) \geq \frac{\varepsilon}{2} \\
1 \cdot f_t(x), & \text{if } -\frac{\varepsilon}{2} \leq \sigma_t \cdot (\langle u_t, x \rangle - y_t) \leq \frac{\varepsilon}{2} \\
\frac{1}{2} \cdot f_t(x), & \text{if } \sigma_t \cdot (\langle u_t, x \rangle - y_t) \leq -\frac{\varepsilon}{2}
\end{cases}$$
Algorithm for $\varepsilon$ – Ball Loss

**$\varepsilon$ – Window Median Algorithm**

Initialize $f_1(x)$: uniform over $B(0,1)$.

For rounds $t = 1, \ldots, T$:

- Observe $u_t$ and query $y_t = \varepsilon – window – median(f_t)$
- Update density:

$$f_{t+1}(x) = \begin{cases} 
\frac{3}{2} \cdot f_t(x), & \text{if } \sigma_t \cdot \langle u_t, x \rangle - y_t \geq \frac{\varepsilon}{2} \\
1 \cdot f_t(x), & \text{if } -\frac{\varepsilon}{2} \leq \sigma_t \cdot \langle u_t, x \rangle - y_t \leq \frac{\varepsilon}{2} \\
\frac{1}{2} \cdot f_t(x), & \text{if } \sigma_t \cdot \langle u_t, x \rangle - y_t \leq -\frac{\varepsilon}{2}
\end{cases}$$

**Main Result**

- $\varepsilon$ – ball loss: $Regret = O(C_0 + d \log 1/\varepsilon)$
Algorithm for $\varepsilon$ – Ball Loss

$\varepsilon$ – Window Median Algorithm

Initialize $f_1(x)$: uniform over $B(0,1)$.

For rounds $t = 1, \ldots, T$:

- Observe $u_t$ and query $y_t = \varepsilon – \text{window} – \text{median}(f_t)$
- Update density:

$$f_{t+1}(x) = \begin{cases} 
\frac{3}{2} \cdot f_t(x), & \text{if } \sigma_t \cdot (\langle u_t, x \rangle - y_t) \geq \frac{\varepsilon}{2} \\
1 \cdot f_t(x), & \text{if } -\frac{\varepsilon}{2} \leq \sigma_t \cdot (\langle u_t, x \rangle - y_t) \leq \frac{\varepsilon}{2} \\
\frac{1}{2} \cdot f_t(x), & \text{if } \sigma_t \cdot (\langle u_t, x \rangle - y_t) \leq -\frac{\varepsilon}{2} 
\end{cases}$$

Proof Idea
Algorithm for $\varepsilon$ – Ball Loss

**$\varepsilon$ – Window Median Algorithm**

Initialize $f_1(x)$: uniform over $B(0,1)$.

For rounds $t = 1, \ldots, T$:

- Observe $u_t$ and query $y_t = \varepsilon - \text{window} - \text{median}(f_t)$
- Update density:

  $$f_{t+1}(x) = \begin{cases} 
  \frac{3}{2} \cdot f_t(x), & \text{if } \sigma_t \cdot (\langle u_t, x \rangle - y_t) \geq \frac{\varepsilon}{2} \\
  1 \cdot f_t(x), & \text{if } -\frac{\varepsilon}{2} \leq \sigma_t \cdot (\langle u_t, x \rangle - y_t) \leq \frac{\varepsilon}{2} \\
  \frac{1}{2} \cdot f_t(x), & \text{if } \sigma_t \cdot (\langle u_t, x \rangle - y_t) \leq -\frac{\varepsilon}{2} 
  \end{cases}$$

**Proof Idea**

1. Given updates above, $f_t(\cdot)$ is always a density.
Algorithm for $\varepsilon$ – Ball Loss

$\varepsilon$ – Window Median Algorithm

Initialize $f_1(x)$: uniform over $B(0,1)$.

For rounds $t = 1, ..., T$:

• Observe $u_t$ and query $y_t = \varepsilon$ – window – median($f_t$)

• Update density:

$$f_{t+1}(x) = \begin{cases} 
\frac{3}{2} \cdot f_t(x), & \text{if } \sigma_t \cdot (\langle u_t, x \rangle - y_t) \geq \frac{\varepsilon}{2} \\
1 \cdot f_t(x), & \text{if } -\frac{\varepsilon}{2} \leq \sigma_t \cdot (\langle u_t, x \rangle - y_t) \leq \frac{\varepsilon}{2} \\
\frac{1}{2} \cdot f_t(x), & \text{if } \sigma_t \cdot (\langle u_t, x \rangle - y_t) \leq -\frac{\varepsilon}{2} 
\end{cases}$$

Proof Idea

1. Given updates above, $f_t(\cdot)$ is always a density.

2. Potential $\Phi_t = \int_{B(\theta^*, \varepsilon/2)} f_t(x) dx$:
Algorithm for $\epsilon$ – Ball Loss

$\epsilon$ – Window Median Algorithm

Initialize $f_1(x)$: uniform over $B(0,1)$.

For rounds $t = 1, \ldots, T$:
- Observe $u_t$ and query $y_t = \epsilon$ – window – median($f_t$)
- Update density:

$$f_{t+1}(x) = \begin{cases} 
\frac{3}{2} \cdot f_t(x), & \text{if } \sigma_t \cdot (\langle u_t, x \rangle - y_t) \geq \frac{\epsilon}{2} \\
1 \cdot f_t(x), & \text{if } -\frac{\epsilon}{2} \leq \sigma_t \cdot (\langle u_t, x \rangle - y_t) \leq \frac{\epsilon}{2} \\
\frac{1}{2} \cdot f_t(x), & \text{if } \sigma_t \cdot (\langle u_t, x \rangle - y_t) \leq -\frac{\epsilon}{2}
\end{cases}$$

Proof Idea

1. Given updates above, $f_t(\cdot)$ is always a density.
2. Potential $\Phi_t = \int_{B(\theta^*, \epsilon/2)} f_t(x) dx$:
   - (weakly) increases in uncorrupted rounds
Algorithm for $\varepsilon$ – Ball Loss

$\varepsilon$ – Window Median Algorithm

Initialize $f_1(x)$: uniform over $B(0,1)$.

For rounds $t = 1, \ldots, T$:

- Observe $u_t$ and query $y_t = \varepsilon – \text{window} – \text{median}(f_t)$
- Update density:

$$f_{t+1}(x) = \begin{cases} 
\frac{3}{2} \cdot f_t(x), & \text{if } \sigma_t \cdot (\langle u_t, x \rangle - y_t) \geq \frac{\varepsilon}{2} \\
1 \cdot f_t(x), & \text{if } -\frac{\varepsilon}{2} \leq \sigma_t \cdot (\langle u_t, x \rangle - y_t) \leq \frac{\varepsilon}{2} \\
\frac{1}{2} \cdot f_t(x), & \text{if } \sigma_t \cdot (\langle u_t, x \rangle - y_t) \leq -\frac{\varepsilon}{2} 
\end{cases}$$

Proof Idea

1. Given updates above, $f_t(\cdot)$ is always a density.

2. Potential $\Phi_t = \int_{B(\theta^*, \varepsilon/2)} f_t(x)dx$:
   - (weakly) increases in uncorrupted rounds
   - decreases by $1/2$ in corrupted ones ($C_0$ in total)
Algorithm for $\epsilon$ – Ball Loss

$\epsilon$ – Window Median Algorithm

Initialize $f_1(x)$: uniform over $B(0,1)$.

For rounds $t = 1, \ldots, T$:

- Observe $u_t$ and query $y_t = \epsilon \text{ – window - median}(f_t)$
- Update density:

$$f_{t+1}(x) = \begin{cases} 
\frac{3}{2} \cdot f_t(x), & \text{if } \sigma_t \cdot (\langle u_t, x \rangle - y_t) \geq \frac{\epsilon}{2} \\
1 \cdot f_t(x), & \text{if } -\frac{\epsilon}{2} \leq \sigma_t \cdot (\langle u_t, x \rangle - y_t) \leq \frac{\epsilon}{2} \\
\frac{1}{2} \cdot f_t(x), & \text{if } \sigma_t \cdot (\langle u_t, x \rangle - y_t) \leq -\frac{\epsilon}{2}
\end{cases}$$

Main Result

- $\epsilon$ – ball loss: $\text{Regret} = O(C_0 + d \log 1/\epsilon)$
Algorithm for $\varepsilon$ – Ball Loss

$\varepsilon$ – Window Median Algorithm

Initialize $f_1(x)$: uniform over $B(0,1)$.

For rounds $t = 1, \ldots, T$:

- Observe $u_t$ and query $y_t = \varepsilon$ – window – median($f_t$)
- Update density:

\[
f_{t+1}(x) = \begin{cases} 
\frac{3}{2} \cdot f_t(x), & \text{if } \sigma_t \cdot (\langle u_t, x \rangle - y_t) \geq \frac{\varepsilon}{2} \\
1 \cdot f_t(x), & \text{if } -\frac{\varepsilon}{2} \leq \sigma_t \cdot (\langle u_t, x \rangle - y_t) \leq \frac{\varepsilon}{2} \\
\frac{1}{2} \cdot f_t(x), & \text{if } \sigma_t \cdot (\langle u_t, x \rangle - y_t) \leq -\frac{\varepsilon}{2}
\end{cases}
\]

Main Result

- $\varepsilon$ – ball loss: $\text{Regret} = O(C_0 + d \log 1/\varepsilon)$
- symmetric loss: $\text{Regret} = O(C_0 + d \log T)$
Algorithm for $\varepsilon$ – Ball Loss

**$\varepsilon$ – Window Median Algorithm**

Initialize $f_1(x)$: uniform over $B(0,1)$.

For rounds $t = 1, \ldots, T$:
- Observe $u_t$ and query $y_t = \varepsilon - \text{window - median}(f_t)$
- Update density:

$$f_{t+1}(x) = \begin{cases} 
\frac{3}{2} \cdot f_t(x), & \text{if } \sigma_t \cdot (\langle u_t, x \rangle - y_t) \geq \frac{\varepsilon}{2} \\
1 \cdot f_t(x), & \text{if } -\frac{\varepsilon}{2} \leq \sigma_t \cdot (\langle u_t, x \rangle - y_t) \leq \frac{\varepsilon}{2} \\
\frac{1}{2} \cdot f_t(x), & \text{if } \sigma_t \cdot (\langle u_t, x \rangle - y_t) \leq -\frac{\varepsilon}{2}
\end{cases}$$

This can happen only $C_0$ times!

**Main Result**

- $\varepsilon$ – ball loss: $\text{Regret} = O(C_0 + d \log \frac{1}{\varepsilon})$
- symmetric loss: $\text{Regret} = O(C_0 + d \log T)$

Runtime $\approx O(T^d \text{ poly}(d,T))$
Efficient Algorithm for Symmetric Loss

Main Result

Polytime algorithm with $\text{Regret} = O(C_1 + d \log T)$ for symmetric loss, where $C_1 = \sum |z_t|$. 
Efficient Algorithm for Symmetric Loss

Main Result

Polytime algorithm with $\text{Regret} = O(C_1 + d \log T)$ for symmetric loss, where $C_1 = \sum_t |z_t|$. 

$C_1 < C_0 = \sum_t 1\{z_t \neq 0\}$
Efficient Algorithm for Symmetric Loss

Main Result

Polytime algorithm with \( \text{Regret} = O(C_1 + d \log T) \) for symmetric loss, where \( C_1 = \sum_t |z_t| \).

Idea

- Maintain “structured” \( f_t(\cdot) \), such that it always is a log-concave density.
- Query centroid of distribution: \( y_t = c g_t = \int x f_t(x) dx \).
- Update: \( f_{t+1}(x) = f_t(x) \cdot \left(1 + \frac{1}{3} \cdot \sigma_t \cdot \langle u_t, x - c g_t \rangle\right) \)
- Finer control over corruptions, as density changes proportionally to how close to \( c g_t \) a point \( x \) is (rather than constant update based on \( \sigma_t \)).
Efficient Algorithm for Symmetric Loss

Main Result

Polytime algorithm with \( \text{Regret} = O(C_1 + d \log T) \) for symmetric loss,
where \( C_1 = \sum_t |z_t| \).

Idea

Maintain “structured” \( f_t(\cdot) \), such that it always is a \textit{log-concave density}.

Query centroid of distribution: \( y_t = c g_t = \int x f_t(x)dx \).

Update: \( f_{t+1}(x) = f_t(x) \cdot \left(1 + \frac{1}{3} \cdot \sigma_t \cdot \langle u_t, x - c g_t \rangle\right) \)

Finer control over corruptions, as density changes proportionally to how close to \( c g_t \) a point \( x \) is (rather than constant update based on \( \sigma_t \)).
Corruption-robust contextual search algorithms with rates:

- $\varepsilon$-ball loss: $\text{Regret} = O(C_0 + d \log 1/\varepsilon)$
- symmetric loss: $\text{Regret} = O(C_1 + d \log T)$, where $C_1 = \sum_t |z_t|$ & polytime
- pricing loss: $\text{Regret} = O(C_0 d^3 \log^3 T)$
Main Result

Corruption-robust contextual search algorithms with rates:

- $\epsilon$-ball loss: $\text{Regret} = O(C_0 + d \log 1/\epsilon)$
- Symmetric loss: $\text{Regret} = O(C_1 + d \log T)$, where $C_1 = \sum_t |z_t|$ & polytime
- Pricing loss: $\text{Regret} = O(C_0 d^3 \log^3 T)$

Open Questions
Main Result

Corruption-robust contextual search algorithms with rates:

- \( \varepsilon \)-ball loss: \( \text{Regret} = O(C_0 + d \log \frac{1}{\varepsilon}) \)
- symmetric loss: \( \text{Regret} = O(C_1 + d \log T) \), where \( C_1 = \sum_t |z_t| \) & polytime
- pricing loss: \( \text{Regret} = O(C_0 d^3 \log^3 T) \)

Open Questions

1. Variant of distribution-based algorithms for pricing loss.
Main Result

Corruption-robust contextual search algorithms with rates:

- $\varepsilon$-ball loss: $\text{Regret} = O(C_0 + d \log 1/\varepsilon)$
- Symmetric loss: $\text{Regret} = O(C_1 + d \log T)$, where $C_1 = \sum_t |z_t|$ & polytime
- Pricing loss: $\text{Regret} = O(C_0 d^3 \log^3 T)$

Open Questions

1. Variant of distribution-based algorithms for pricing loss.
2. Algorithms with $\text{Regret} = O(C_1 + d \log d)$ for symmetric loss.
Main Result

Corruption-robust contextual search algorithms with rates:

- \( \varepsilon \)-ball loss: \( \text{Regret} = O(C_0 + d \log 1/\varepsilon) \)
- symmetric loss: \( \text{Regret} = O(C_1 + d \log T) \), where \( C_1 = \sum_t |z_t| \) & polytime
- pricing loss: \( \text{Regret} = O(C_0 d^3 \log^3 T) \)

Open Questions

1. Variant of distribution-based algorithms for pricing loss.
2. Algorithms with \( \text{Regret} = O(C_1 + d \log d) \) for symmetric loss.
3. Polytime algorithm for \( \varepsilon \)-ball loss.
Corruption-robust contextual search algorithms with rates:

- $\varepsilon$–ball loss: $\text{Regret} = O(C_0 + d \log \frac{1}{\varepsilon})$
- Symmetric loss: $\text{Regret} = O(C_1 + d \log T)$, where $C_1 = \sum_t |z_t|$ & polytime
- Pricing loss: $\text{Regret} = O(C_0 d^3 \log^3 T)$

Open Questions

1. Variant of distribution-based algorithms for pricing loss.
2. Algorithms with $\text{Regret} = O(C_1 + d \log d)$ for symmetric loss.
3. Polytime algorithm for $\varepsilon$–ball loss.

Thank you!