Corruption-Robust Contextual Search

Chara Podimata

(UC Berkeley → MIT)

Based on joint works with

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**]

Contextual Search Realizable Version

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**]

Contextual Search Realizable Version

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

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**]

Contextual Search Realizable Version

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 $\mathrm{sign}(y_t - \langle u_t, \theta^\star \rangle) \in \{-1, +1\}$ (binary feedback).

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

Contextual Search Realizable Version

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

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

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 $\mathrm{sign}(y_t - \langle u_t, \theta^\star \rangle) \in \{-1, +1\}$ (binary feedback).

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

Contextual Search Realizable Version

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

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

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 $\mathrm{sign}(y_t - \langle u_t, \theta^\star \rangle) \in \{-1, +1\}$ (binary feedback).
4. Learner incurs **(but does not observe)** loss $\ell(y_t, \langle u_t, \theta^\star \rangle) \in [0,1]$.

[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 $\theta^\star \in \mathbb{R}^d$ same across rounds.

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

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 $\text{sign}(y_t - \langle u_t, \theta^\star \rangle) \in \{-1, +1\}$ (binary feedback).
4. Learner incurs **(but does not observe)** loss $\boxed{\ell(y_t, \langle u_t, \theta^\star \rangle) \in [0,1].}$

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

[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 $\theta^\star \in \mathbb{R}^d$ same across rounds.

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

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 $\mathrm{sign}(y_t - \langle u_t, \theta^\star \rangle) \in \{-1, +1\}$ (binary feedback).
4. Learner incurs **(but does not observe)** loss $\boxed{\ell(y_t, \langle u_t, \theta^\star \rangle) \in [0,1].}$

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

$Regret$ = 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**]

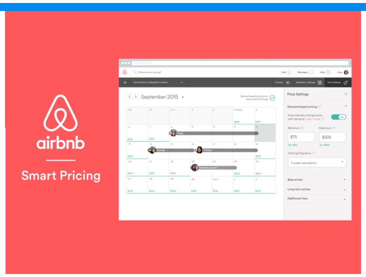
Contextual Search Realizable Version

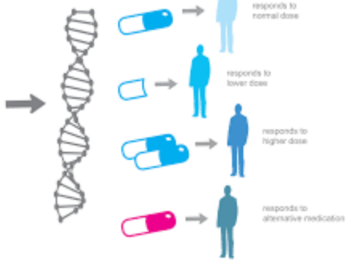
"Realizable" = Exists hidden $\theta^\star \in \mathbb{R}^d$ same across rounds.

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

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 $\mathrm{sign}(y_t - \langle u_t, \theta^\star \rangle) \in \{-1, +1\}$ (binary feedback).
4. Learner incurs **(but does not observe)** loss $\boxed{\ell(y_t, \langle u_t, \theta^\star \rangle) \in [0,1].}$

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





$Regret$ = 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**]

Contextual Search Realizable Version

Known Results

Loss	$\ell(y_t, y_t^\star)$	Lower Bound [R]	Upper Bound [R]

[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

Known Results

Loss	$\ell(y_t, y_t^\star)$	Lower Bound [R]	Upper Bound [R]
ε −ball	$1\{\lvert y_t - y_t^\star\rvert \geq \varepsilon\}$	$\Omega(d \log 1/\varepsilon)$	$O(d \log 1/\varepsilon)$ **[LLV17]**

[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

Known Results

Loss	$\ell(y_t, y_t^\star)$	Lower Bound [R]	Upper Bound [R]
ε −ball	$1\{\lvert y_t - y_t^\star \rvert \geq \varepsilon\}$	$\Omega(d \log 1/\varepsilon)$	$O(d \log 1/\varepsilon)$ [LLV17]
symmetric	$\lvert y_t - y_t^\star \rvert$	$\Omega(d)$	$O(d \log d)$ [LLS21]

[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

Known Results

Loss	$\ell(y_t, y_t^\star)$	Lower Bound [R]	Upper Bound [R]
ε –ball	$1\{\lvert y_t - y_t^\star\rvert \geq \varepsilon\}$	$\Omega(d \log 1/\varepsilon)$	$O(d \log 1/\varepsilon)$ [LLV17]
symmetric	$\lvert y_t - y_t^\star\rvert$	$\Omega(d)$	$O(d \log d)$ [LLS21]
pricing	$y_t^\star - y_t \cdot 1\{y_t \leq y_t^\star\}$	$\Omega(d \log\log T)$	$O(d \log\log T + d \log d)$ [LLS21]

[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 $\theta^\star \in \mathbb{R}^d$ same across rounds.

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

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 $\sigma_t = \mathrm{sign}(y_t - y_t^\star) \in \{-1, +1\}$ (binary feedback).
4. Learner incurs **(but does not observe)** loss $\ell(y_t, y_t^\star) \in [0,1]$.

Contextual Search with Adversarial Noise

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

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

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 $\sigma_t = \text{sign}(y_t - y_t^\star) \in \{-1, +1\}$ (binary feedback).
4. Learner incurs **(but does not observe)** loss $\ell(y_t, y_t^\star) \in [0,1]$.

$$y_t^\star = \langle u_t, \theta^\star \rangle$$

Contextual Search with Adversarial Noise

Exists hidden $\theta^\star \in \mathbb{R}^d$ 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 $\sigma_t = \mathrm{sign}(y_t - y_t^\star) \in \{-1, +1\}$ (binary feedback).
4. Learner incurs **(but does not observe)** loss $\ell(y_t, y_t^\star) \in [0,1]$.

$y_t^\star = \langle u_t, \theta^\star \rangle + z_t$
$z_t \in [-1,1]$: adaptively and adversarially chosen

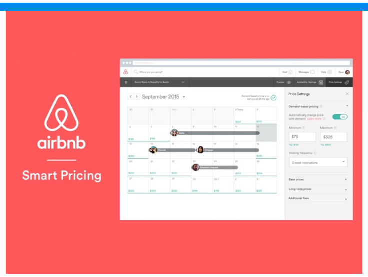
Contextual Search with Adversarial Noise

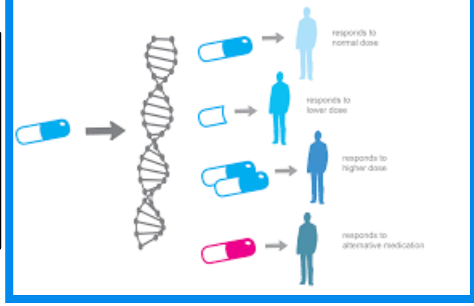
Exists hidden $\theta^\star \in \mathbb{R}^d$ same across rounds.

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 $\sigma_t = \text{sign}(y_t - y_t^\star) \in \{-1, +1\}$ (binary feedback).
4. Learner incurs **(but does not observe)** loss $\ell(y_t, y_t^\star) \in [0,1]$.

$y_t^\star = \langle u_t, \theta^\star \rangle + z_t$
$z_t \in [-1,1]$: adaptively and adversarially chosen





Contextual Search with Adversarial Noise

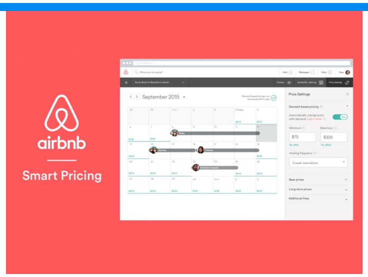
Exists hidden $\theta^\star \in \mathbb{R}^d$ same across rounds.

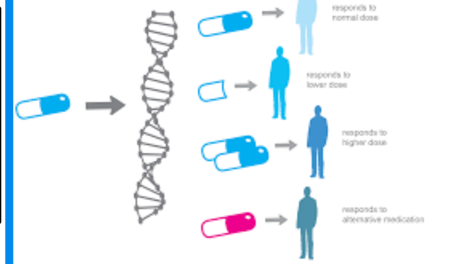
For rounds $t = 1, \dots, T$:

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 $\sigma_t = \text{sign}(y_t - y_t^\star) \in \{-1, +1\}$ (binary feedback).
4. Learner incurs **(but does not observe)** loss $\ell(y_t, y_t^\star) \in [0,1]$.

$y_t^\star = \langle u_t, \theta^\star \rangle + z_t$

$z_t \in [-1,1]$: adaptively and adversarially chosen





Desiderata for our Algorithms

✓ Graceful degradation of regret with C

✓ No knowledge of C assumed (*agnostic*)

Main Results

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

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

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

[Paes Leme, **P.**, Schneider, **COLT22**]

ε –ball loss: $Regret = O(C_0 + d \log 1/\varepsilon)$

symmetric loss: polytime, $Regret = O(C_1 + d \log T), where\ C_1 = \sum_t |z_t|$

Contextual Search Realizable Version

Known Results

Loss	$\ell(y_t, y_t^\star)$	Lower Bound [R]	Upper Bound [R]
ε −ball	$1\{\lvert y_t - y_t^\star\rvert \geq \varepsilon\}$	$\Omega(d \log 1/\varepsilon)$	$O(d \log 1/\varepsilon)$ [LLV17]
symmetric	$\lvert y_t - y_t^\star\rvert$	$\Omega(d)$	$O(d \log d)$ [LLS21]
pricing	$y_t^\star - y_t \cdot 1\{y_t \leq y_t^\star\}$	$\Omega(d \log\log T)$	$O(d \log\log T + d \log d)$ [LLS21]

[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 $\theta^\star \in \mathbb{R}^d$ same across rounds.

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

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 $\sigma_t = \text{sign}(y_t - y_t^\star) \in \{-1, +1\}$ (binary feedback).
4. Learner incurs **(but does not observe)** loss $\ell(y_t, y_t^\star) \in [0,1]$.

$y_t^\star = \langle u_t, \theta^\star \rangle + z_t$
$z_t \in [-1,1]$: adaptively and adversarially chosen

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

for $z_t \in \{0,1\}$: $C_0 = \#\{z_t = 1\}$

- $Regret = O(C_0\, d^3 \log^3 T)$ for pricing, symmetric loss
- $Regret = O(C_0\, d^3 \log^3 1/\varepsilon)$ for ε –ball loss

Runtime
$poly\,(d, \log T)^{poly(\log T)}$

Traditional Approach for Contextual Search Algorithms

[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

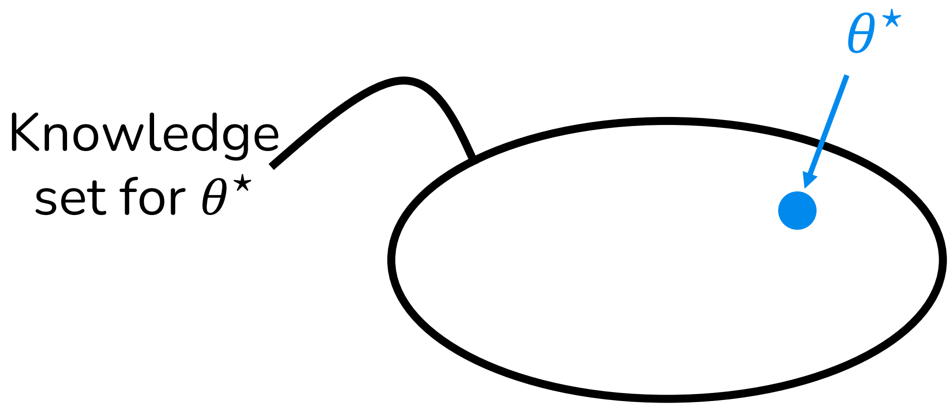
For rounds $t = 1, \ldots, T$:

[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 $t = 1, \dots, T$:

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



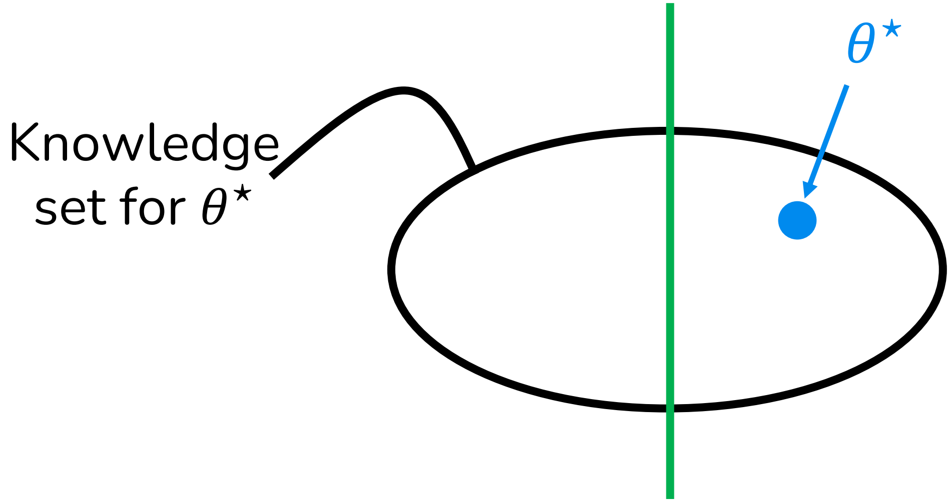
[Cohen, Lobel, Paes Leme, **EC16/MS19**], [Lobel, Paes Leme, Vladu, **EC17/OR18**], [Paes Leme, Schneider, **FOCS18**], [Liu, Paes Leme, Schneider, **SODA21**]

Traditional Approach for Contextual Search Algorithms

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

- Maintain active **knowledge set** with feasible values for $\theta^\star$.
- Learner chooses y_t to make enough ”**progress**”

  (e.g., $y_t = \langle u_t, centroid\ of\ knowledge\ set \rangle$).



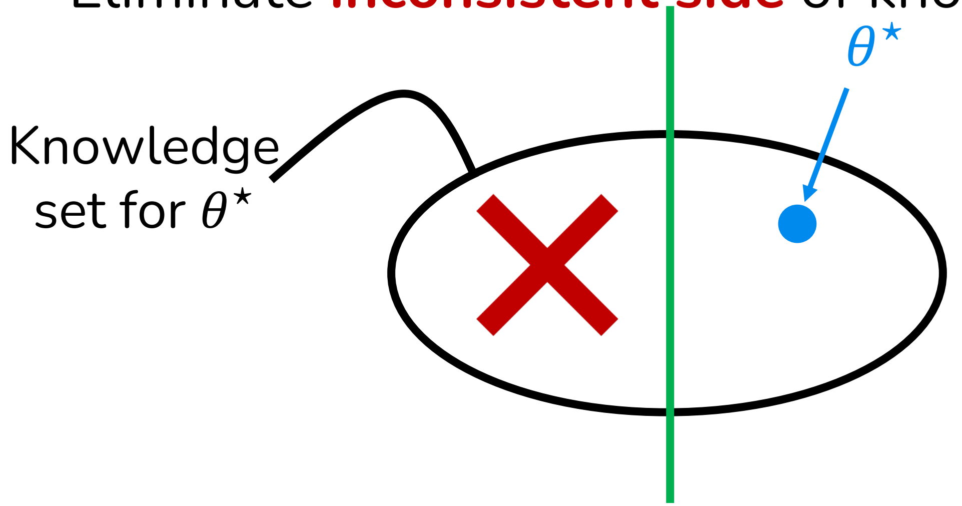
[Cohen, Lobel, Paes Leme, **EC16/MS19**], [Lobel, Paes Leme, Vladu, **EC17/OR18**], [Paes Leme, Schneider, **FOCS18**], [Liu, Paes Leme, Schneider, **SODA21**]

Traditional Approach for Contextual Search Algorithms

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

- Maintain active **knowledge set** with feasible values for $\theta^\star$.
- Learner chooses y_t to make enough "**progress**"

  (e.g., $y_t = \langle u_t, centroid\ of\ knowledge\ set \rangle$).
- Eliminate **inconsistent side** of knowledge set.



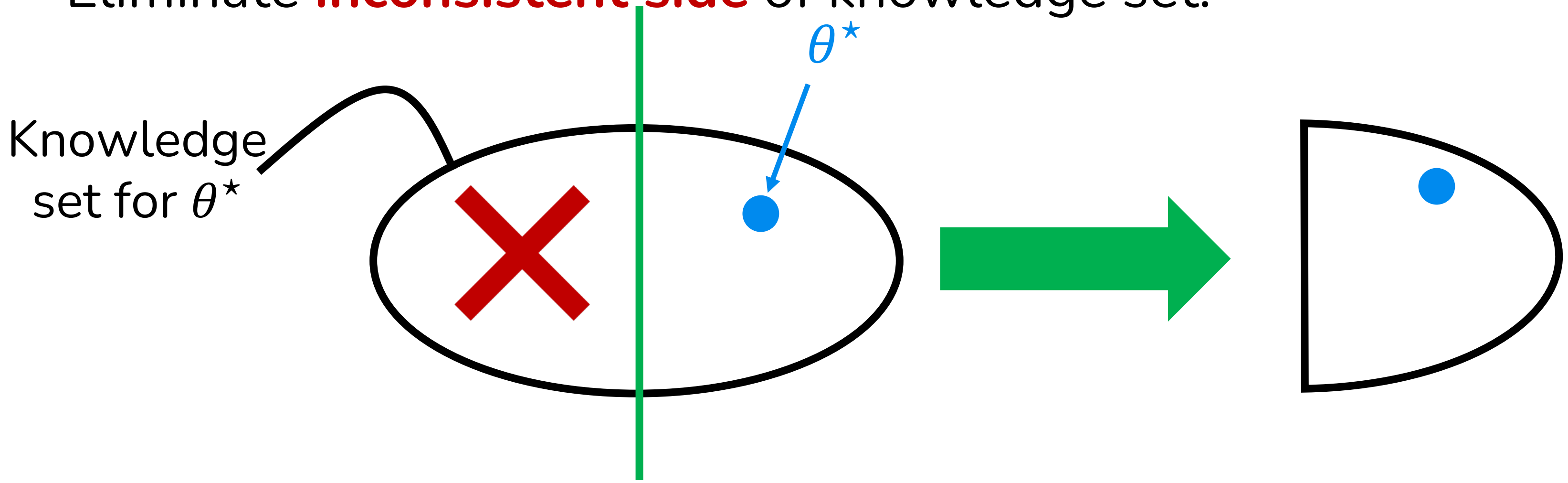
[Cohen, Lobel, Paes Leme, **EC16/MS19**], [Lobel, Paes Leme, Vladu, **EC17/OR18**], [Paes Leme, Schneider, **FOCS18**], [Liu, Paes Leme, Schneider, **SODA21**]

Traditional Approach for Contextual Search Algorithms

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

- Maintain active **knowledge set** with feasible values for $\theta^\star$.
- Learner chooses y_t to make enough "**progress**"

  (e.g., $y_t = \langle u_t, centroid\ of\ knowledge\ set \rangle$).
- Eliminate **inconsistent side** of knowledge set.



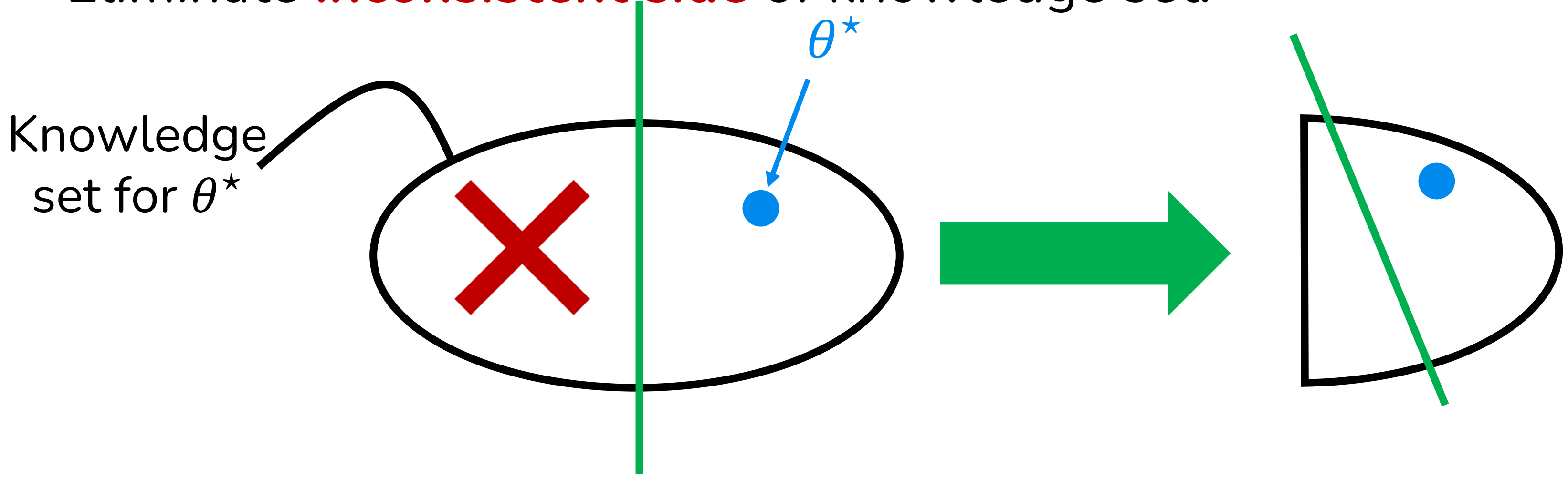
[Cohen, Lobel, Paes Leme, **EC16/MS19**], [Lobel, Paes Leme, Vladu, **EC17/OR18**], [Paes Leme, Schneider, **FOCS18**], [Liu, Paes Leme, Schneider, **SODA21**]

Traditional Approach for Contextual Search Algorithms

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

- Maintain active **knowledge set** with feasible values for $\theta^\star$.
- Learner chooses y_t to make enough "**progress**"

  (e.g., $y_t = \langle u_t, centroid\ of\ knowledge\ set \rangle$).
- Eliminate **inconsistent side** of knowledge set.



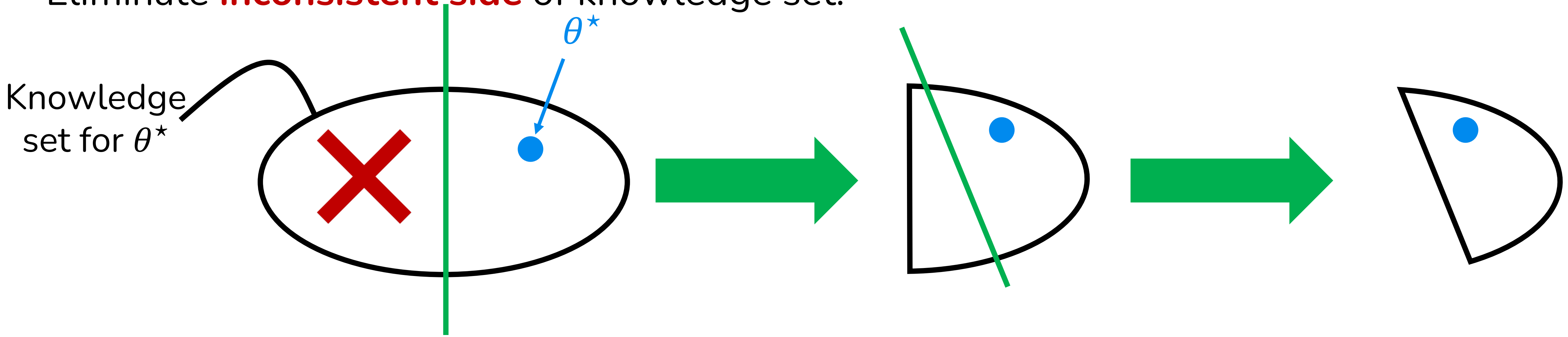
[Cohen, Lobel, Paes Leme, **EC16/MS19**], [Lobel, Paes Leme, Vladu, **EC17/OR18**], [Paes Leme, Schneider, **FOCS18**], [Liu, Paes Leme, Schneider, **SODA21**]

Traditional Approach for Contextual Search Algorithms

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

- Maintain active **knowledge set** with feasible values for $\theta^\star$.
- Learner chooses y_t to make enough ”**progress**”

  (e.g., $y_t = \langle u_t, centroid\ of\ knowledge\ set \rangle$).
- Eliminate **inconsistent side** of knowledge set.



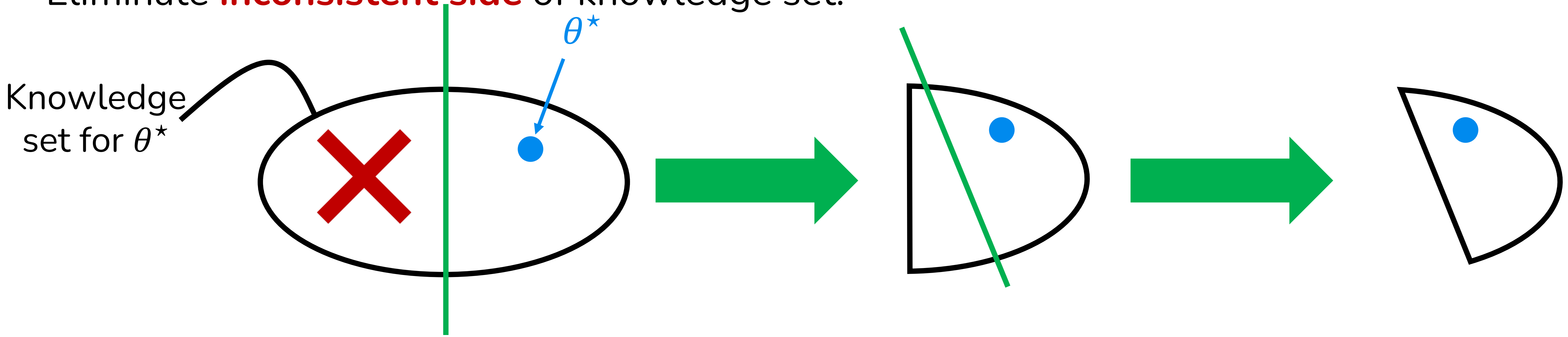
[Cohen, Lobel, Paes Leme, **EC16/MS19**], [Lobel, Paes Leme, Vladu, **EC17/OR18**], [Paes Leme, Schneider, **FOCS18**], [Liu, Paes Leme, Schneider, **SODA21**]

Traditional Approach for Contextual Search Algorithms

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

- Maintain active **knowledge set** with feasible values for $\theta^\star$.
- Learner chooses y_t to make enough ”**progress**”

  (e.g., $y_t = \langle u_t, centroid\ of\ knowledge\ set \rangle$).
- Eliminate **inconsistent side** of knowledge set.



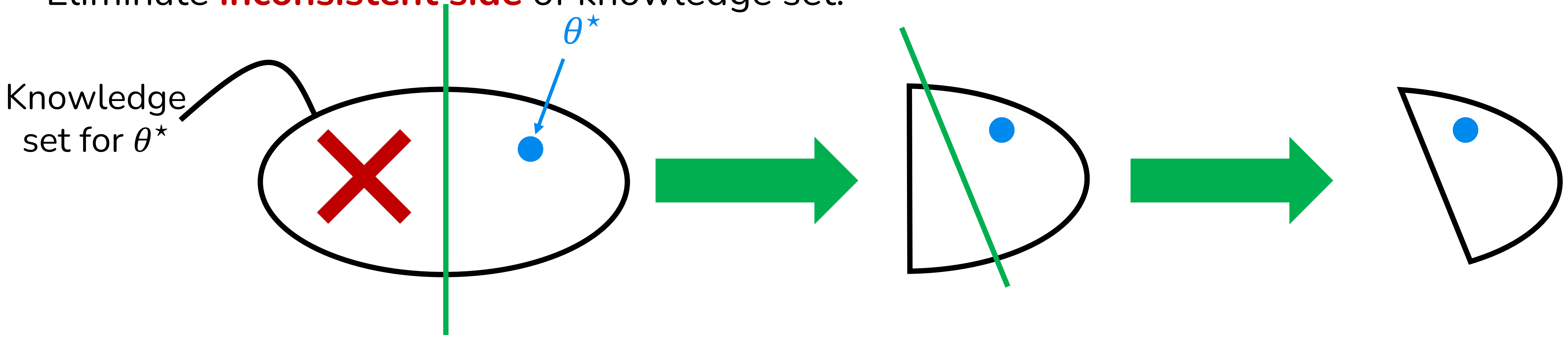
Aggressively introducing cuts → **fast**, **logarithmic** bounds

[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 $t = 1, \dots, T$:

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



Aggressively introducing cuts → **fast**, **logarithmic** bounds

$Regret \leq O(d \cdot \log T)$

[Lobel, Paes Leme, Vladu, **EC17/OR18**]

Traditional Approach for Contextual Search Algorithms

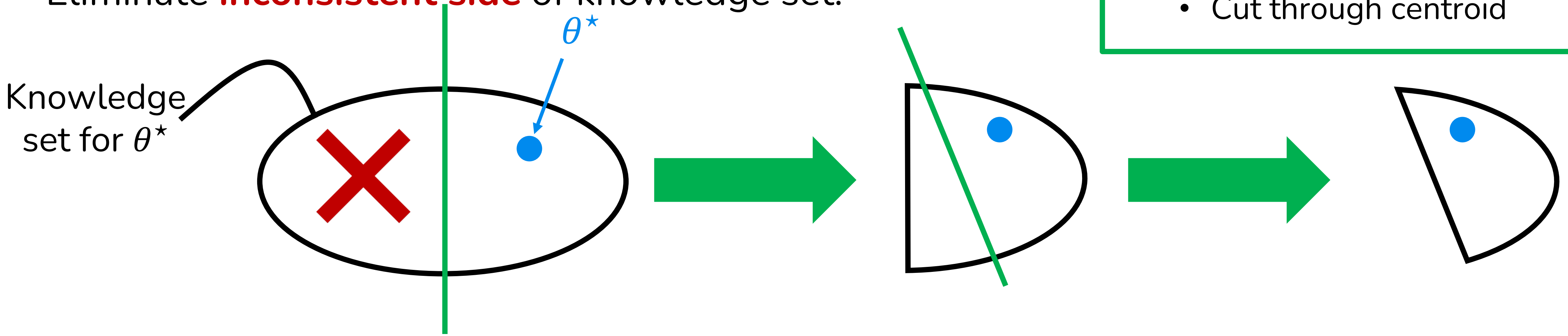
For rounds $t = 1, \dots, T$:

- Maintain active **knowledge set** with feasible values for $\theta^\star$.
- Learner chooses y_t to make enough **"progress"**

  (e.g., $y_t = \langle u_t, centroid\ of\ knowledge\ set \rangle$).
- Eliminate **inconsistent side** of knowledge set.

Important properties of cut

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



Aggressively introducing cuts → **fast**, **logarithmic** bounds

$Regret \leq O(d \cdot \log T)$

[Lobel, Paes Leme, Vladu, **EC17/OR18**]

Robustify-ing Knowledge-Set-Based Techniques

Idea Overview

1. Create a robust version of the knowledge-set-based algorithm that is robust to a **known** amount of corruption $\bar{c} \approx \log T$.
2. **Unknown** C: Run a variant of the Multi-Layering Race technique from [Lykouris, Mirrokni, Paes Leme, **STOC18**].

Robustify-ing Knowledge-Set-Based Techniques

Idea Overview

1. Create a robust version of the knowledge-set-based algorithm that is robust to a **known** amount of corruption $\bar{c} \approx \log T$.
2. **Unknown** C: Run a variant of the Multi-Layering Race technique from [Lykouris, Mirrokni, Paes Leme, **STOC18**].

Robust Volumetric Progress

Known $\bar{c}$

Robust Volumetric Progress

Known $\bar{c}$

Challenge 1

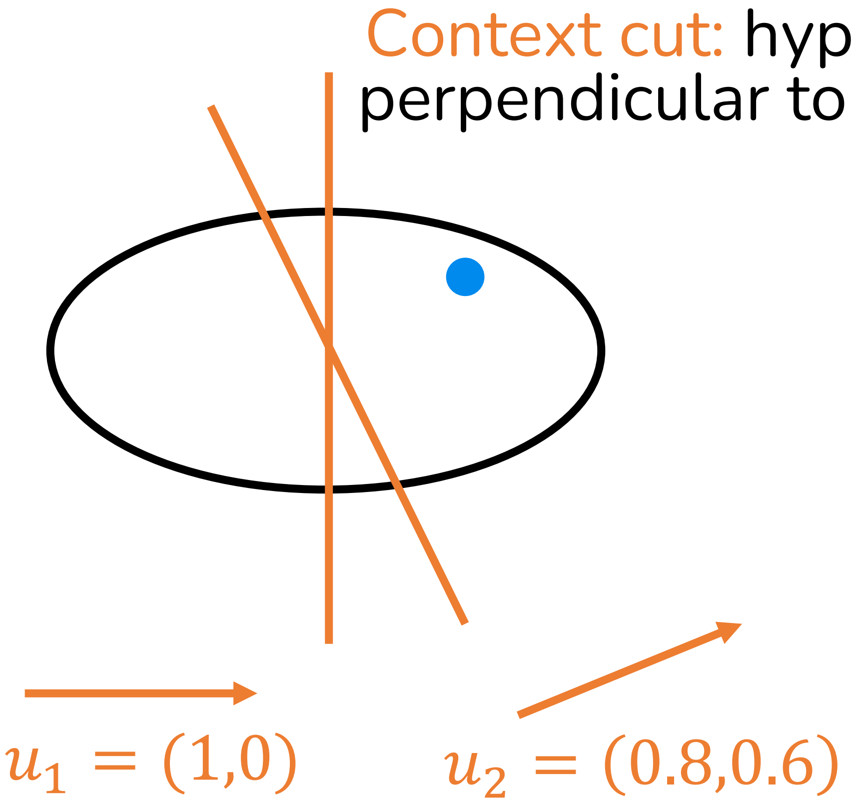
We cannot repeat the same query (contexts are different at different rounds).

Robust Volumetric Progress

Known $\bar{c}$

Challenge 1

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



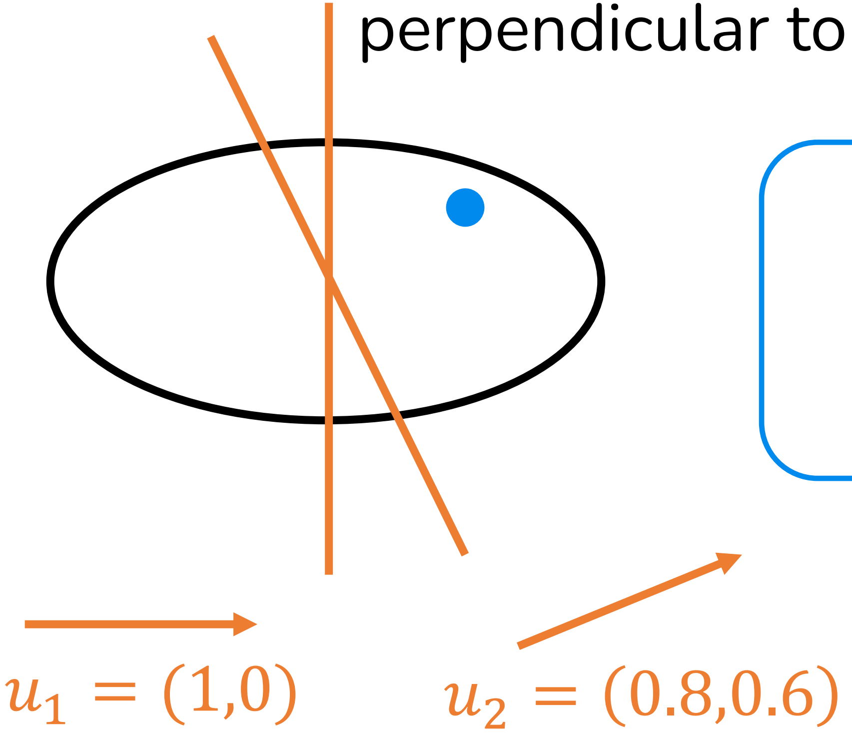
Robust Volumetric Progress

Known $\bar{c}$

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" $\leq \bar{c}$.

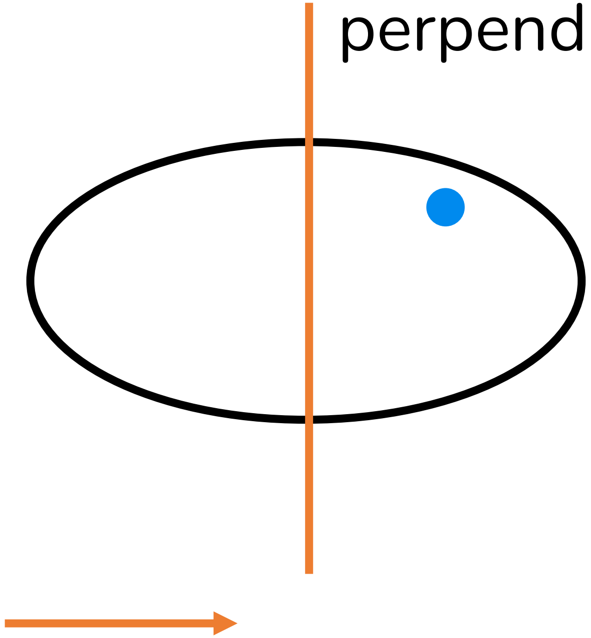
Robust Volumetric Progress

Known $\bar{c}$

Challenge 1

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

Context cut: hyperplane perpendicular to context



$u_1 = (1,0)$

Idea 1

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

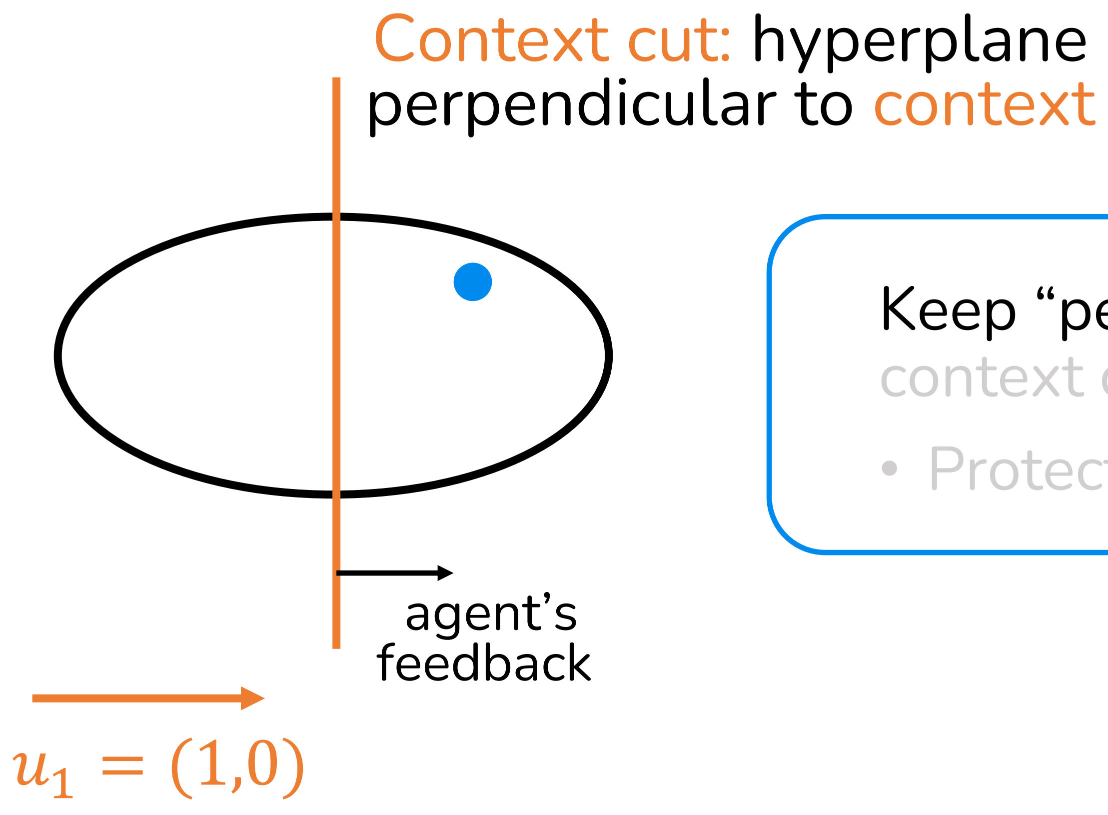
- Protected region: all parameters with “penalty” $\leq \bar{c}$.

Robust Volumetric Progress

Known $\bar{c}$

Challenge 1

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



Idea 1

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

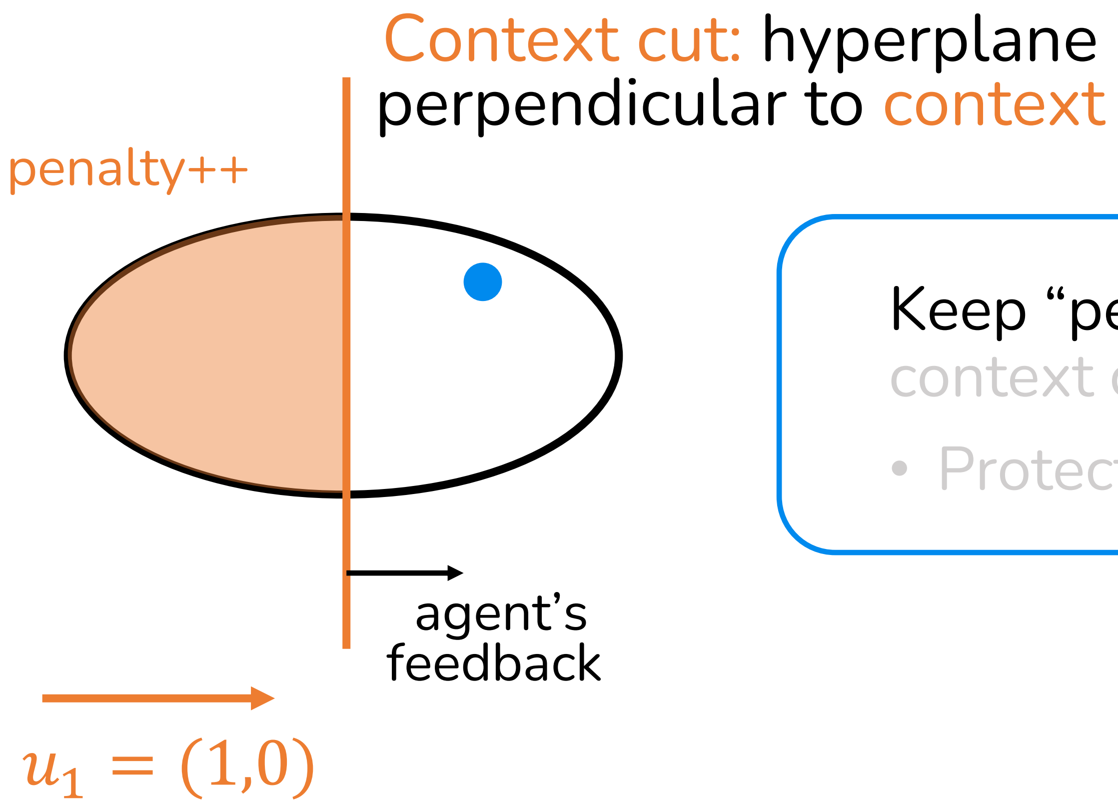
- Protected region: all parameters with "penalty" $\leq \bar{c}$.

Robust Volumetric Progress

Known $\bar{c}$

Challenge 1

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



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" $\leq \bar{c}$.

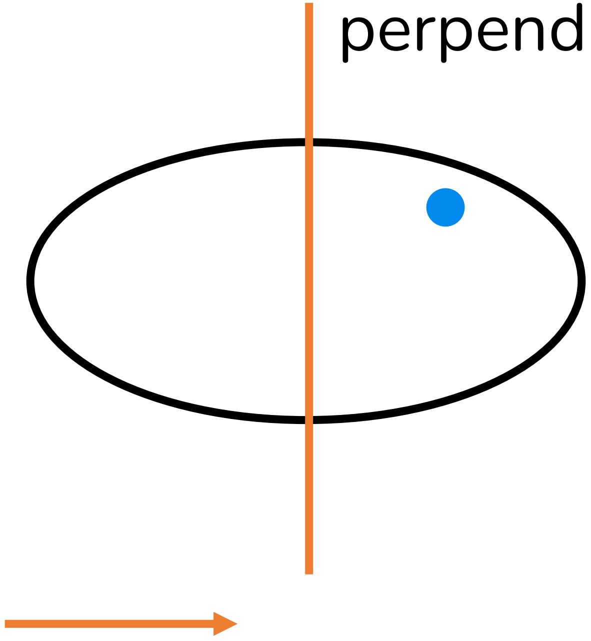
Robust Volumetric Progress

Known $\bar{c}$

Challenge 1

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

Context cut: hyperplane perpendicular to context



$u_1 = (1,0)$

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” $\leq \bar{c}$.

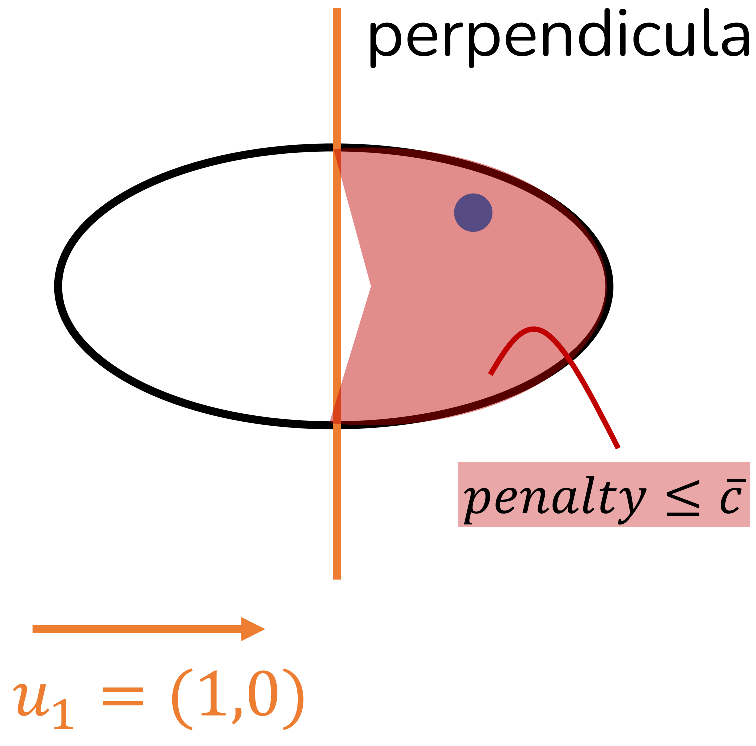
Robust Volumetric Progress

Known $\bar{c}$

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" $\leq \bar{c}$.

Robust Volumetric Progress

Known $\bar{c}$

Robust Volumetric Progress

Known $\bar{c}$

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 $\bar{\boldsymbol{c}} = \mathbf{1}$, 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

Known $\bar{c}$

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.

Important Properties of Valid Cut

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

Robust Volumetric Progress

Known $\bar{c}$

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 $\theta^\star$
   - Retains protected region on one side.
2. Volumetric progress
   - Cross **close** to centroid.

Robust Volumetric Progress

Known $\bar{c}$

Challenge 2

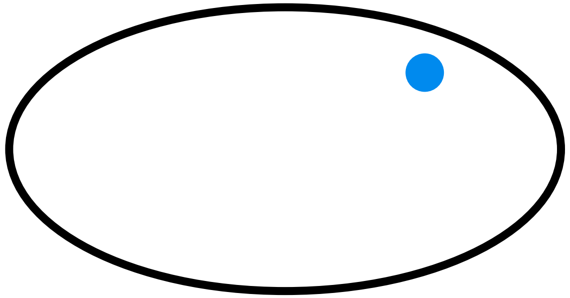
We may never have a context cut with the protected region fully on one side.

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 $\bar{\boldsymbol{c}} = \mathbf{1}$, no such context cut.

Important Properties of Valid Cut

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

Known $\bar{c}$

Robust Volumetric Progress

Challenge 2

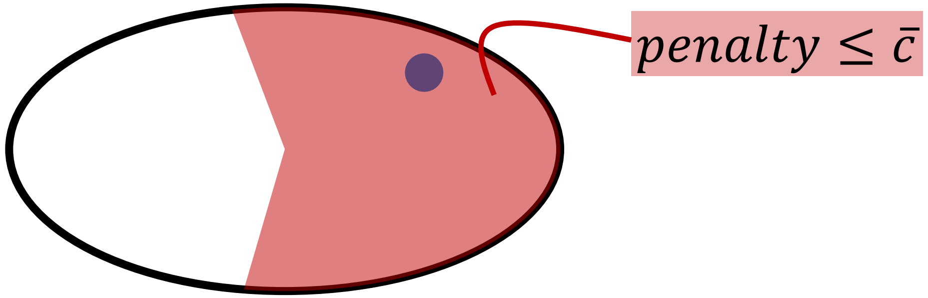
We may never have a context cut with the protected region fully on one side.

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 $\theta^\star$
   - Retains protected region on one side.
2. Volumetric progress
   - Cross **close** to centroid.

Robust Volumetric Progress

Challenge 2

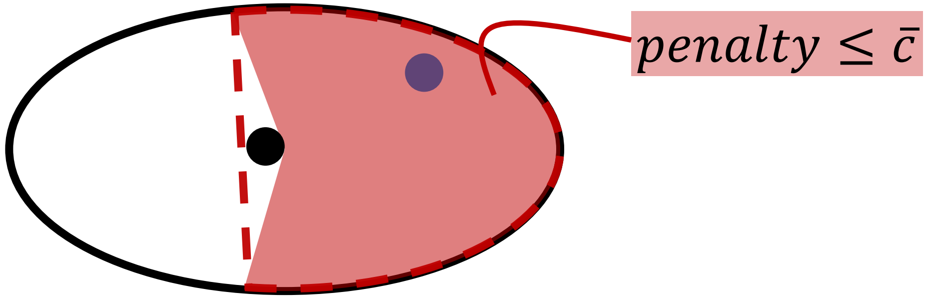
We may never have a context cut with the protected region fully on one side.

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 $\theta^\star$
   - Retains protected region on one side.
2. Volumetric progress
   - Cross **close** to centroid.

Known $\bar{c}$

Robust Volumetric Progress

Challenge 2

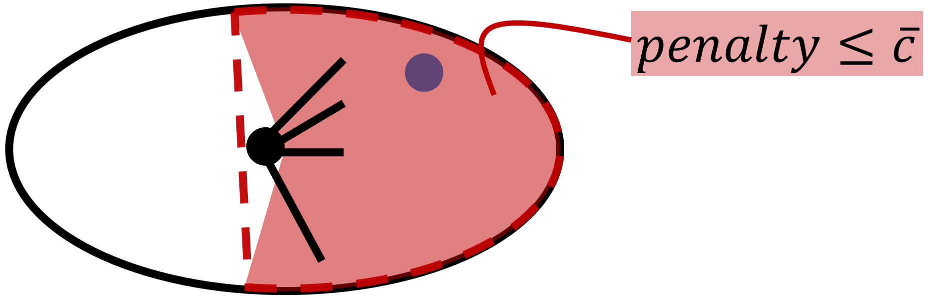
We may never have a context cut with the protected region fully on one side.

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 $\bar{\boldsymbol{c}} = \mathbf{1}$, no such context cut.

Important Properties of Valid Cut

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

Robust Volumetric Progress

Known $\bar{c}$

Challenge 2

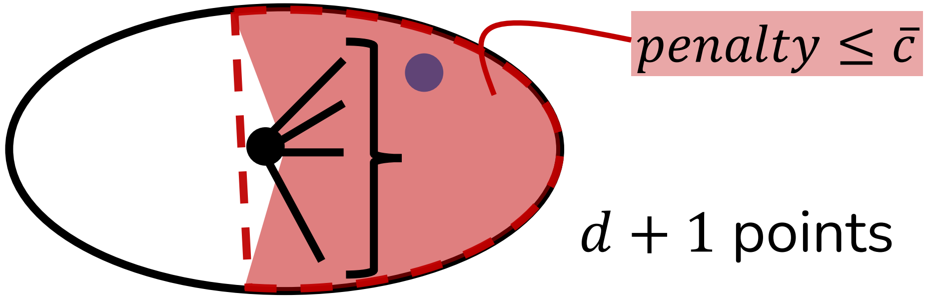
We may never have a context cut with the protected region fully on one side.

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 $\theta^\star$
   - Retains protected region on one side.
2. Volumetric progress
   - Cross **close** to centroid.

Robust Volumetric Progress

Known $\bar{c}$

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".

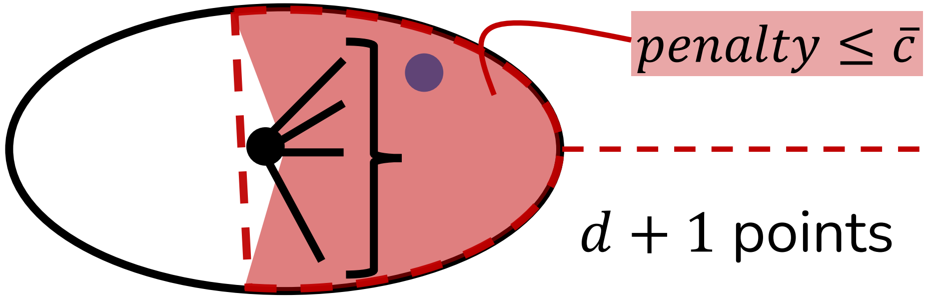
Idea 3

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 $\boldsymbol{\bar{c} = 1}$, no such context cut.

Important Properties of Valid Cut

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



Each penalty for black point
→ attributed to ≥ 1 protected point
→ $penalty(black) \leq \bar{c} \cdot (d+1)$

Robust Volumetric Progress

Known $\bar{c}$

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".

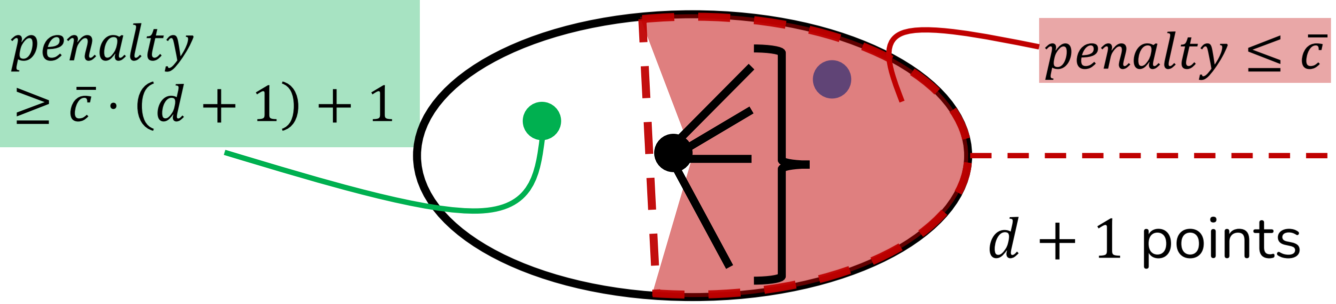
Idea 3

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 $\boldsymbol{\bar{c} = 1}$, no such context cut.

Important Properties of Valid Cut

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



Each penalty for black point
→ attributed to ≥ 1 protected point
→ $penalty(black) \leq \bar{c} \cdot (d+1)$

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

Idea 4

Use Perceptron to find a valid cut.

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

Important Properties of Valid Cut

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

A Fundamentally Different Approach

A Fundamentally Different Approach

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

A Fundamentally Different Approach

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

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

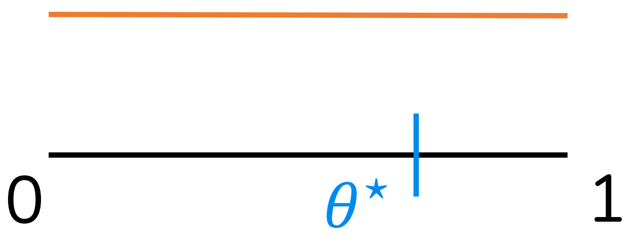
→ Higher weight to more probable values.

A Fundamentally Different Approach

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

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

→ Higher weight to more probable values.

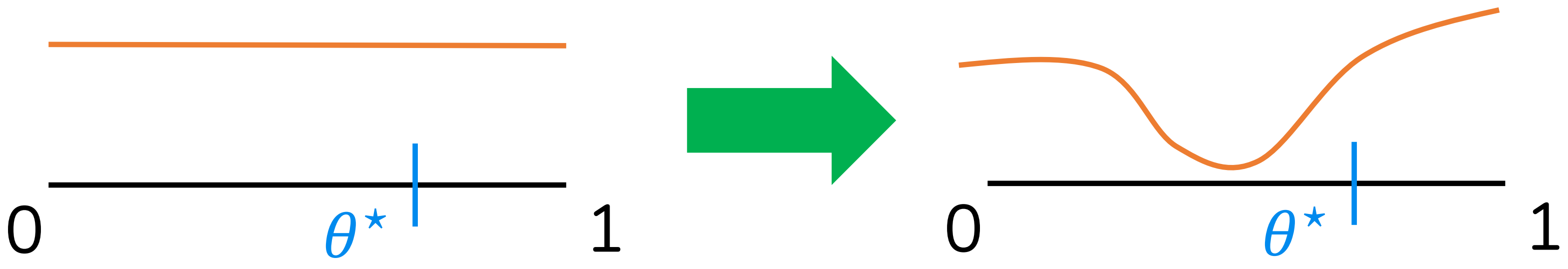


A Fundamentally Different Approach

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

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

→ Higher weight to more probable values.

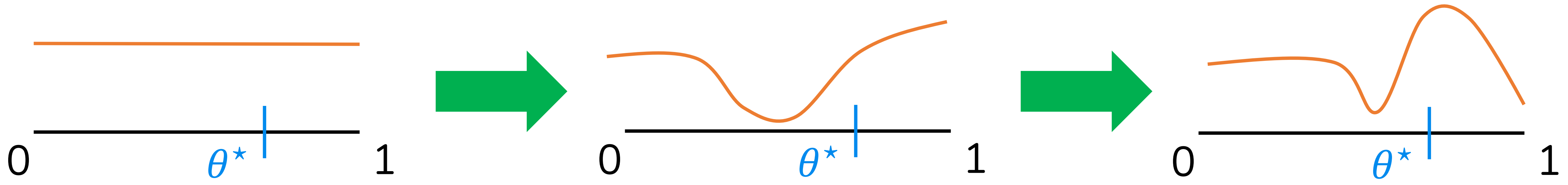


A Fundamentally Different Approach

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

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

→ Higher weight to more probable values.



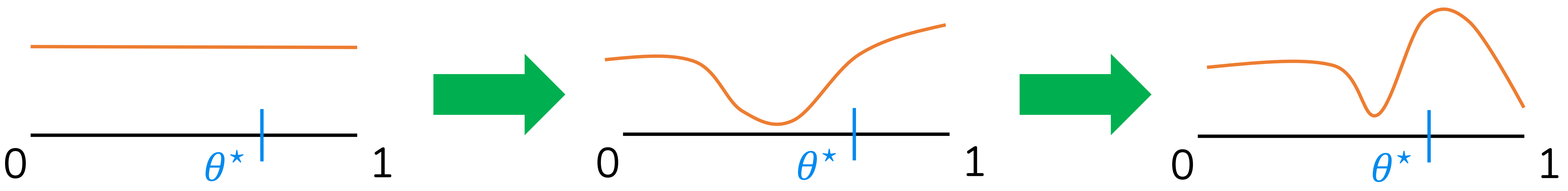
A Fundamentally Different Approach

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

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

→ Higher weight to more probable values.

Seemingly more "forgiving" approach → faster bounds for corruption-robust

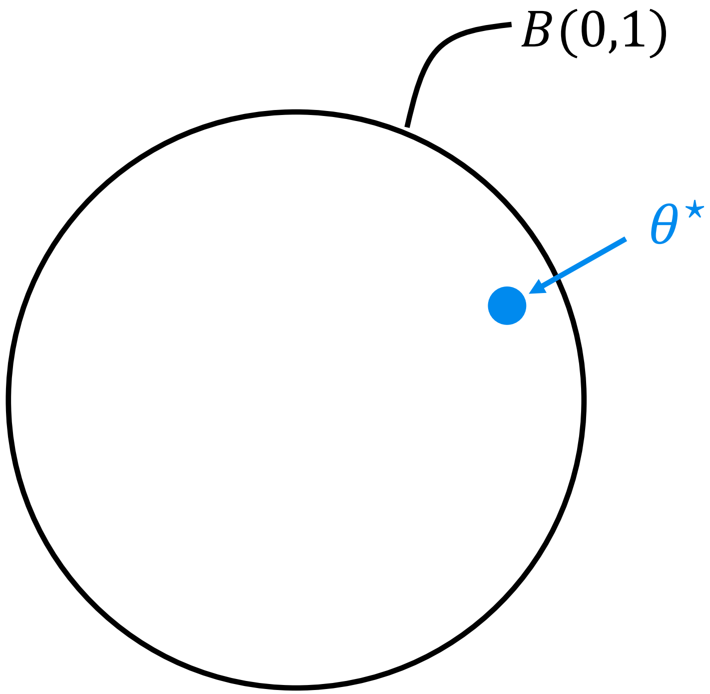


Algorithm for $\varepsilon-$ Ball Loss

Algorithm for ε – Ball Loss

$\boldsymbol{\varepsilon}$ – Window Median Algorithm

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

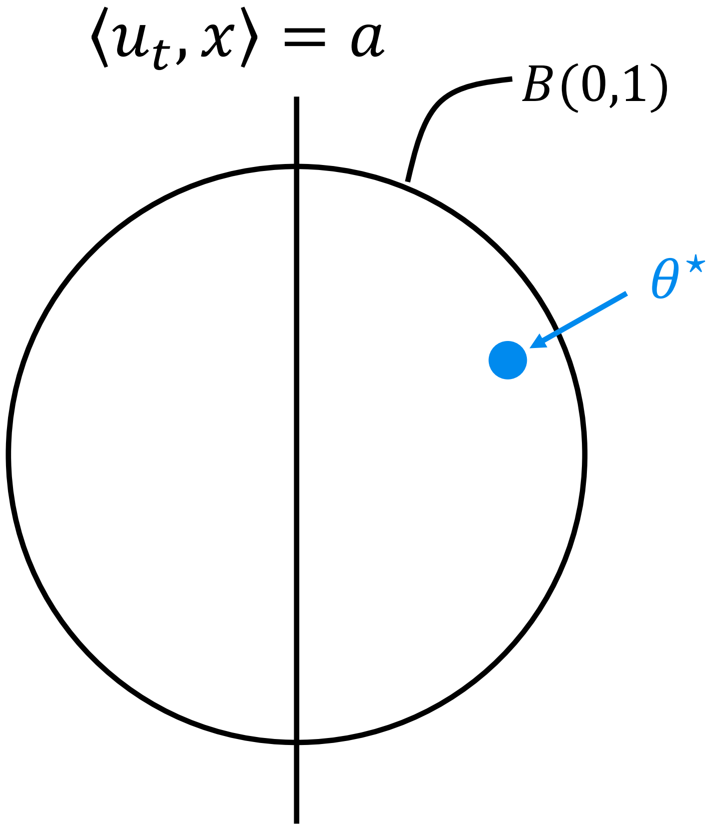


Algorithm for ε − Ball Loss

$\boldsymbol{\varepsilon}$ − Window Median Algorithm

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

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



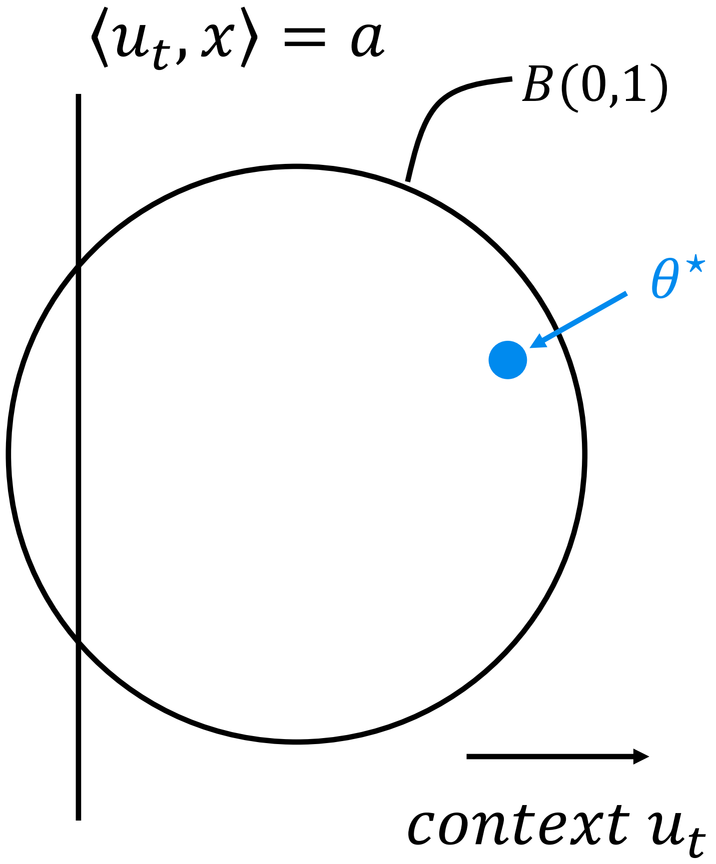
Algorithm for ε – Ball Loss

$\boldsymbol{\varepsilon}$ – Window Median Algorithm

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

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

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



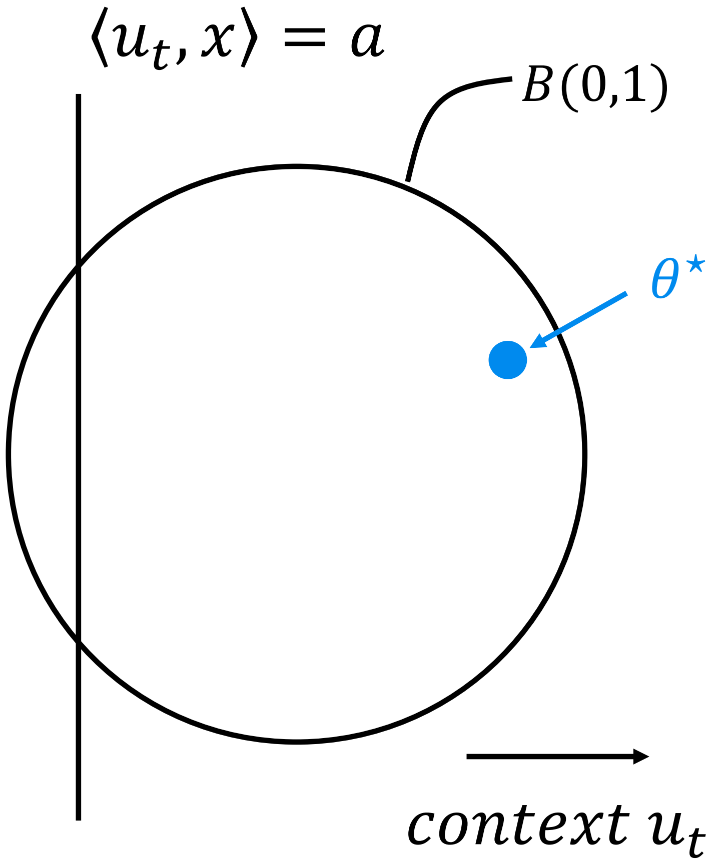
Algorithm for ε – Ball Loss

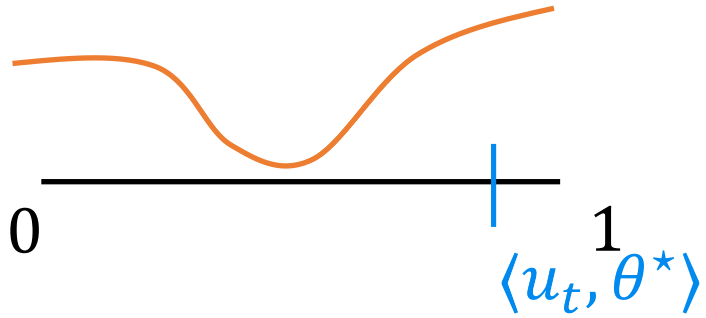
$\boldsymbol{\varepsilon}$ – Window Median Algorithm

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

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

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





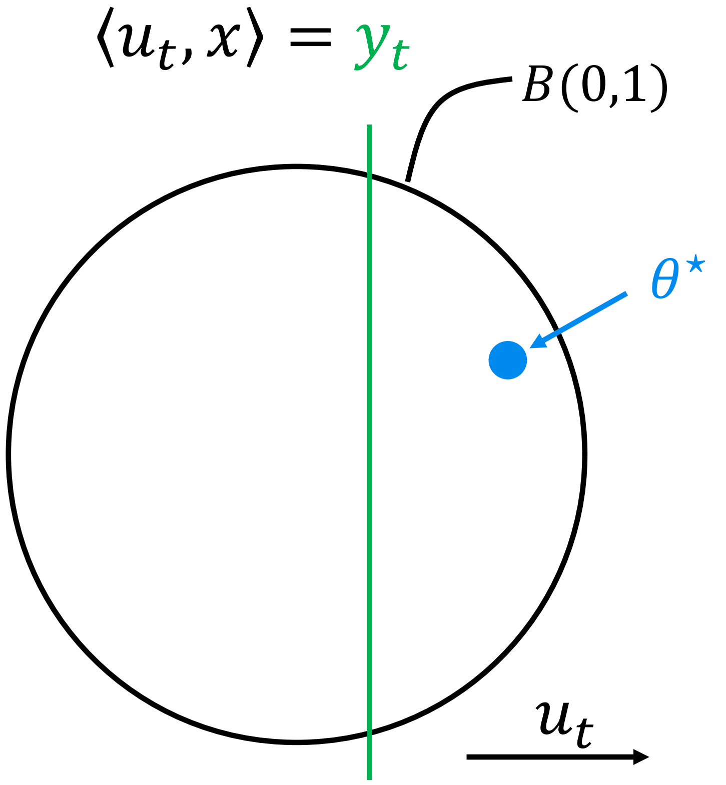
Algorithm for ε – Ball Loss

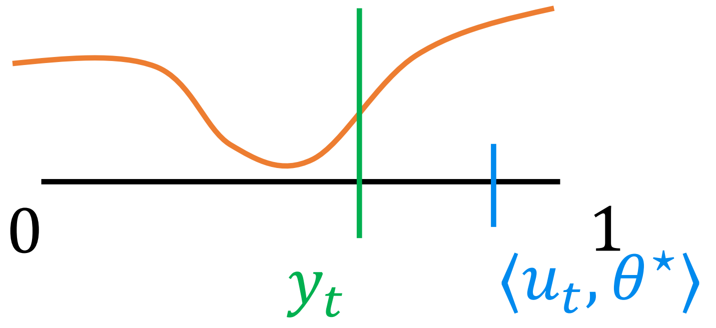
$\boldsymbol{\varepsilon}$ – Window Median Algorithm

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

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

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





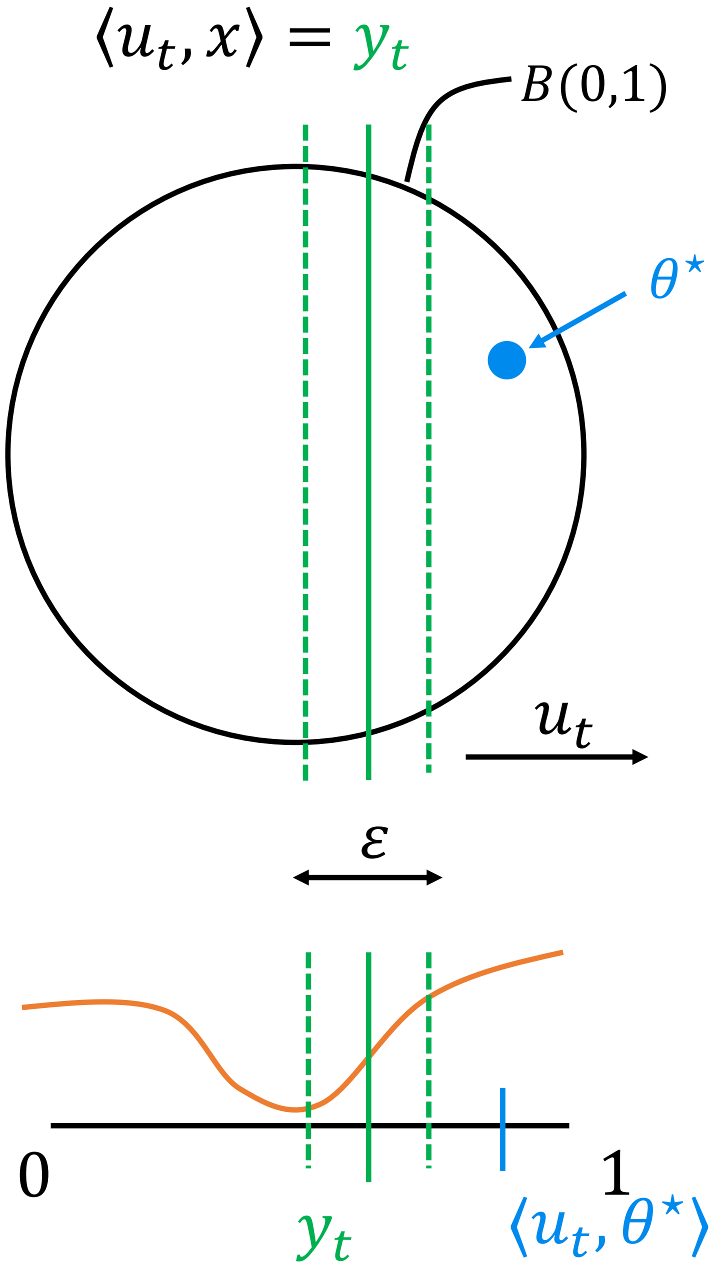
Algorithm for ε – Ball Loss

$\boldsymbol{\varepsilon}$ – Window Median Algorithm

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

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

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



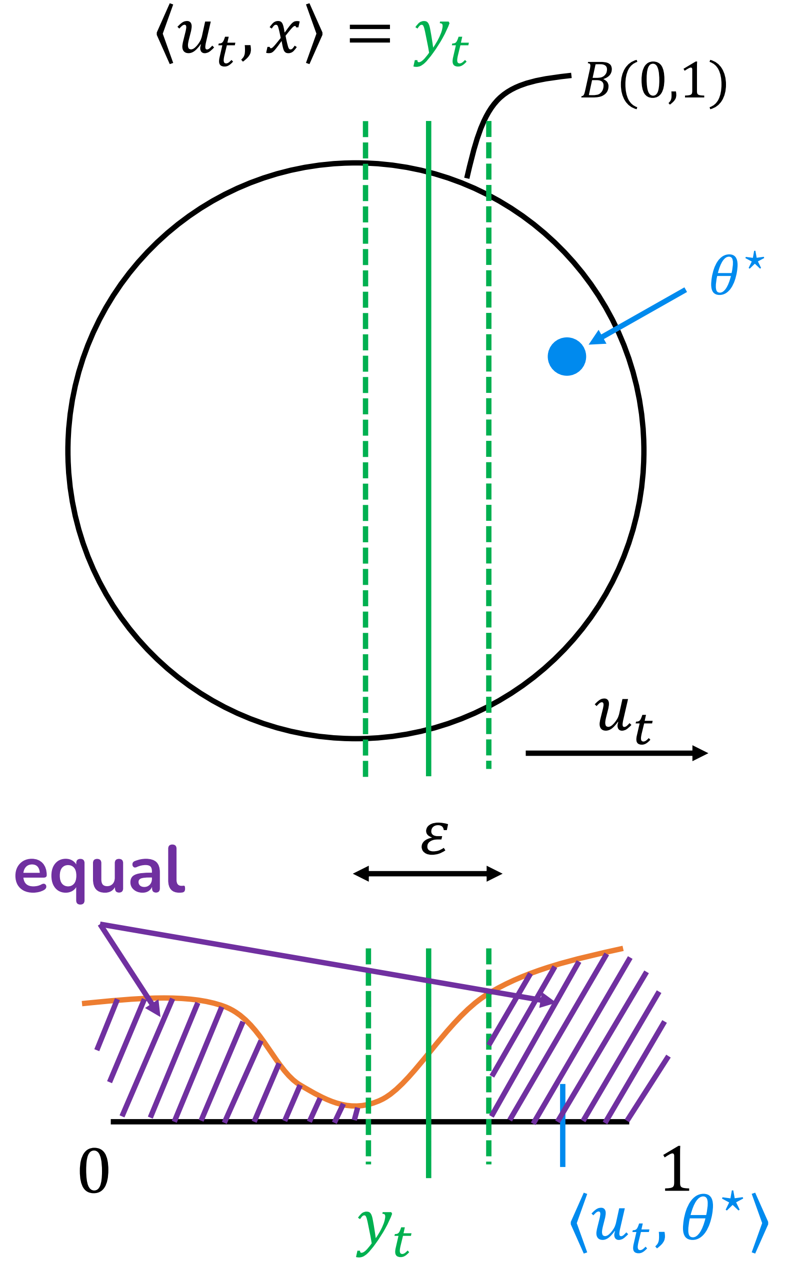
Algorithm for ε – Ball Loss

$\boldsymbol{\varepsilon}$ – Window Median Algorithm

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

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

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



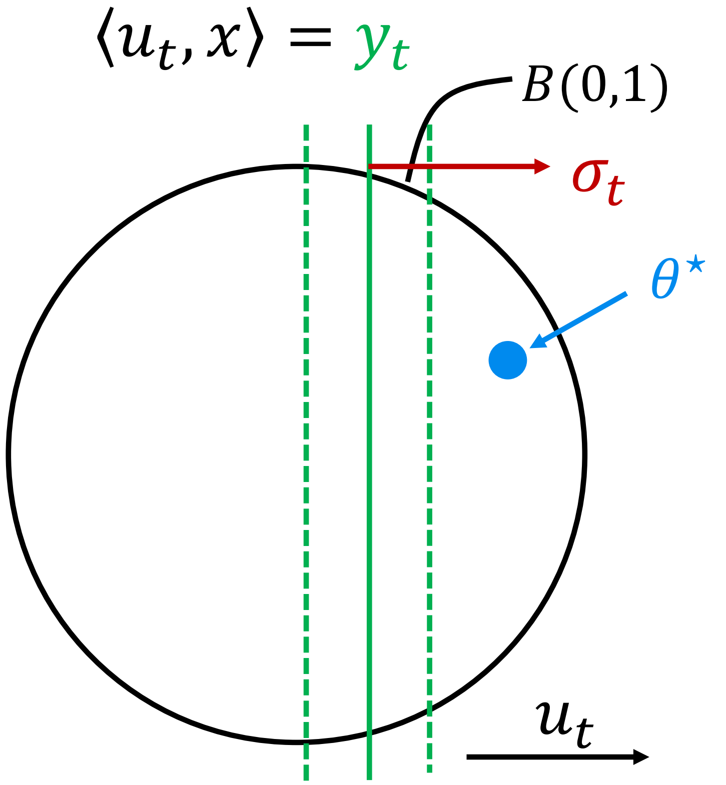
Algorithm for ε – Ball Loss

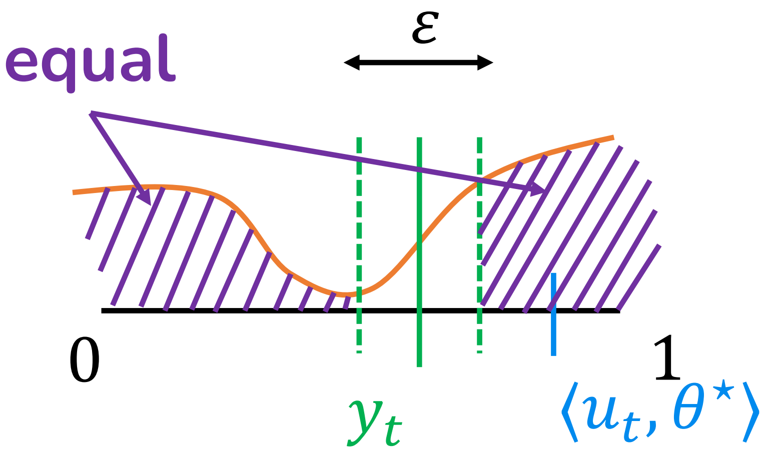
$\boldsymbol{\varepsilon}$ – Window Median Algorithm

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 ε – Ball Loss

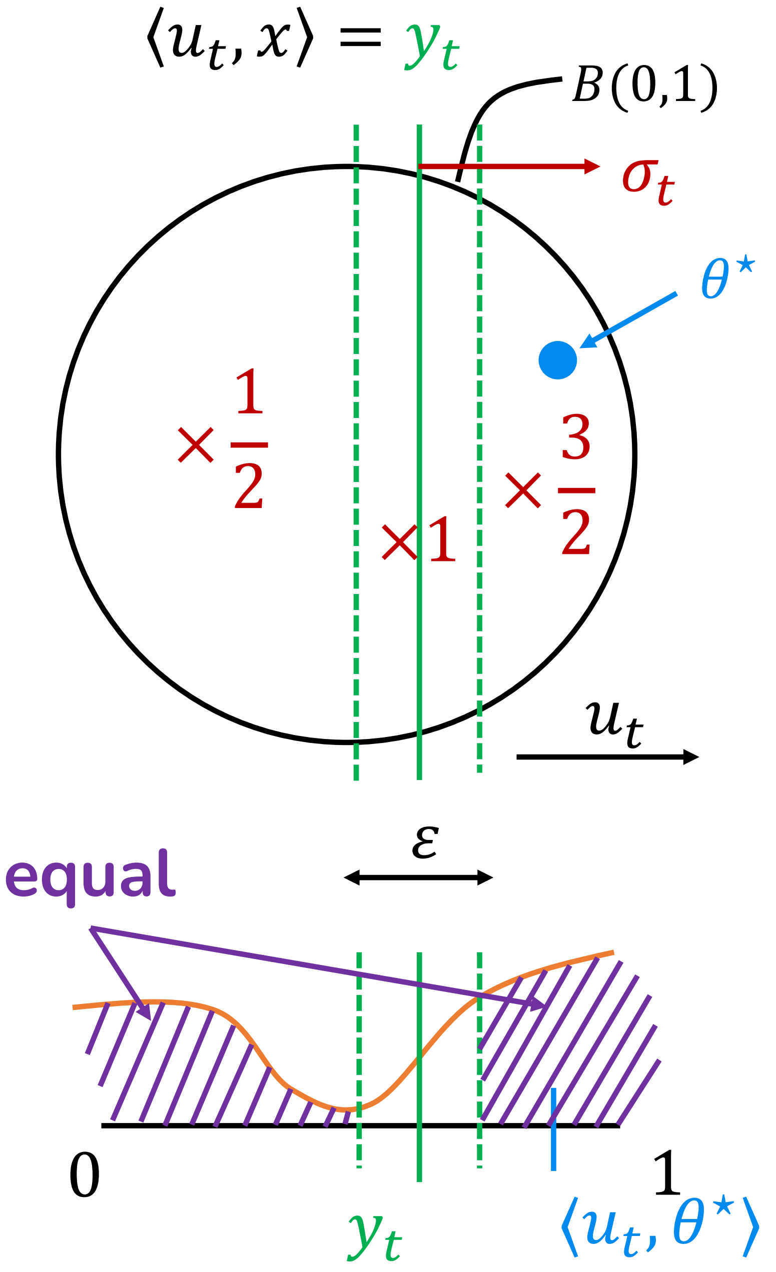
$\boldsymbol{\varepsilon}$ – Window Median Algorithm

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

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

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

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



Algorithm for ε – Ball Loss

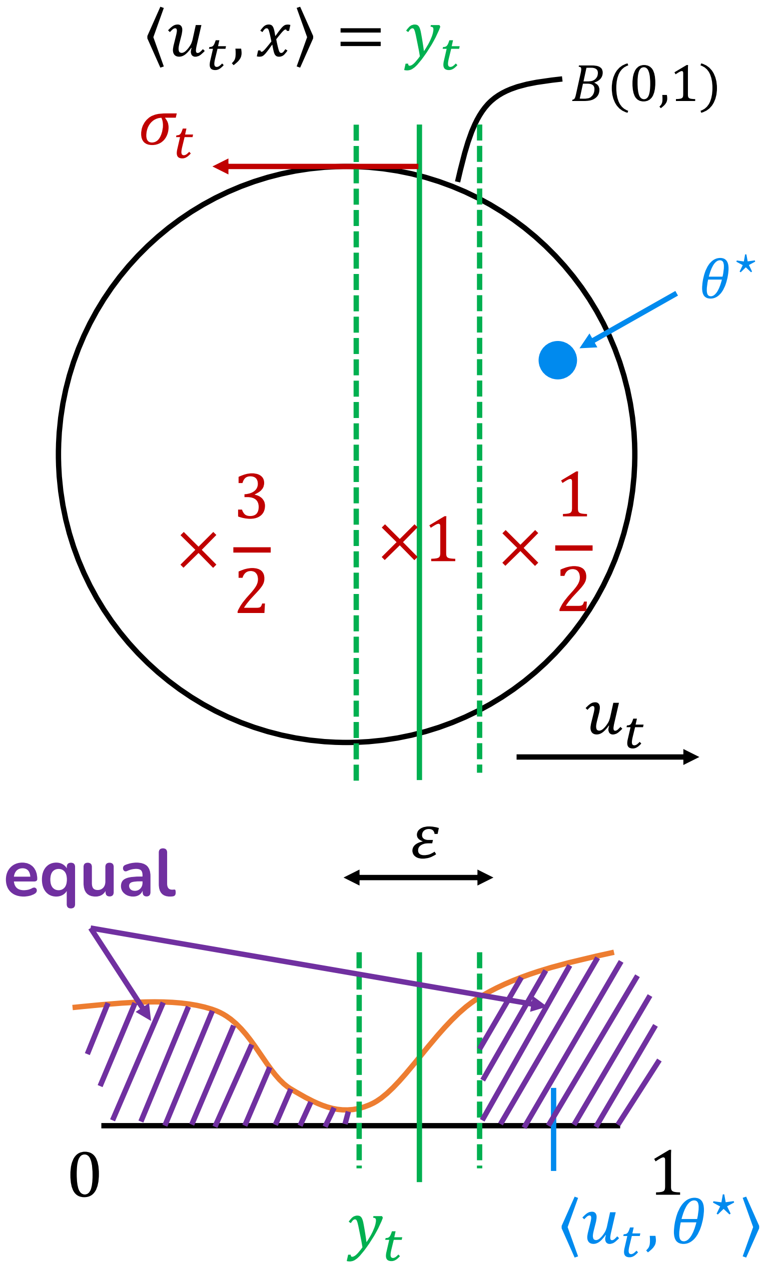
ε – Window Median Algorithm

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

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

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

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



Algorithm for ε – Ball Loss

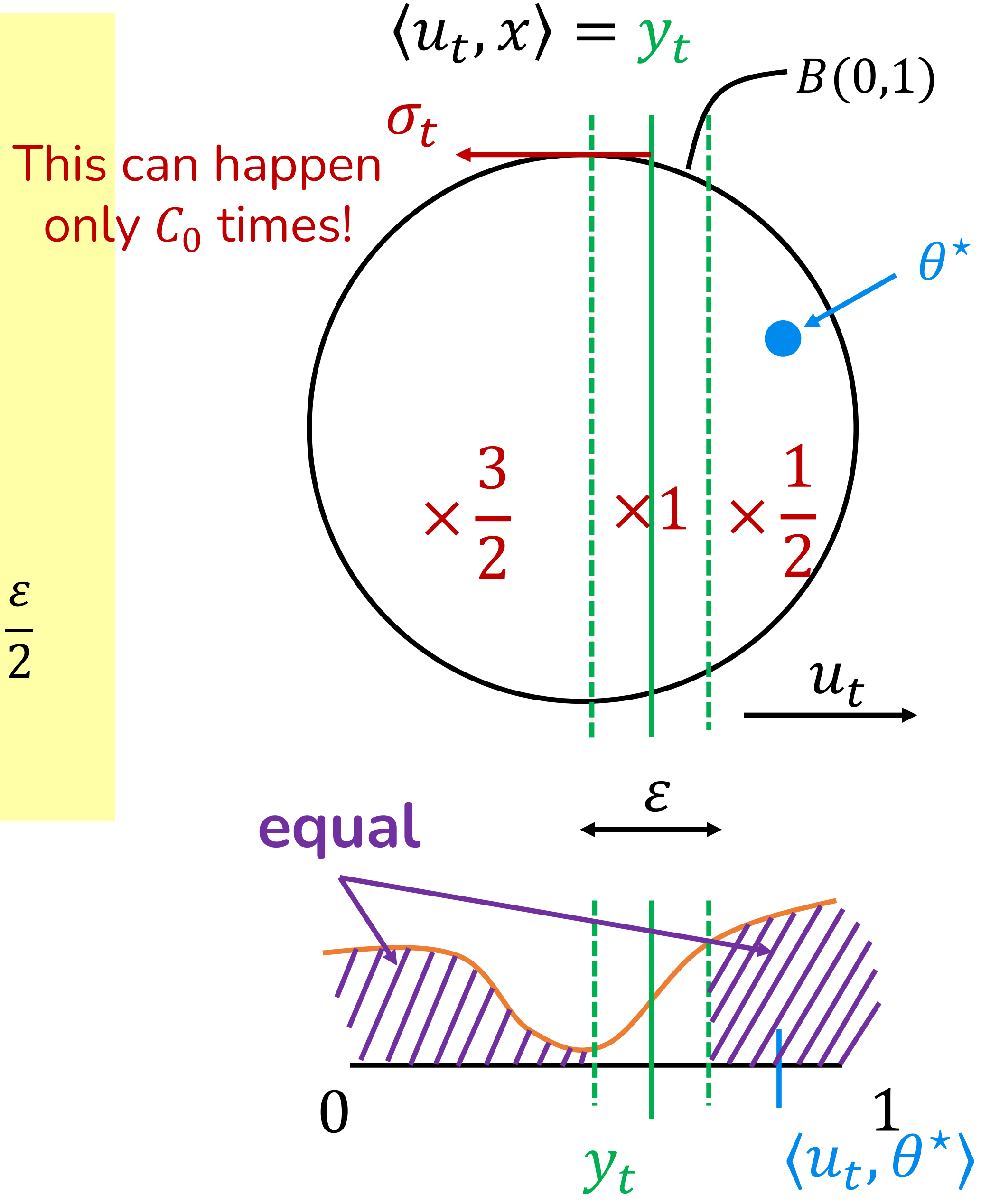
ε – Window Median Algorithm

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:

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



Algorithm for ε – Ball Loss

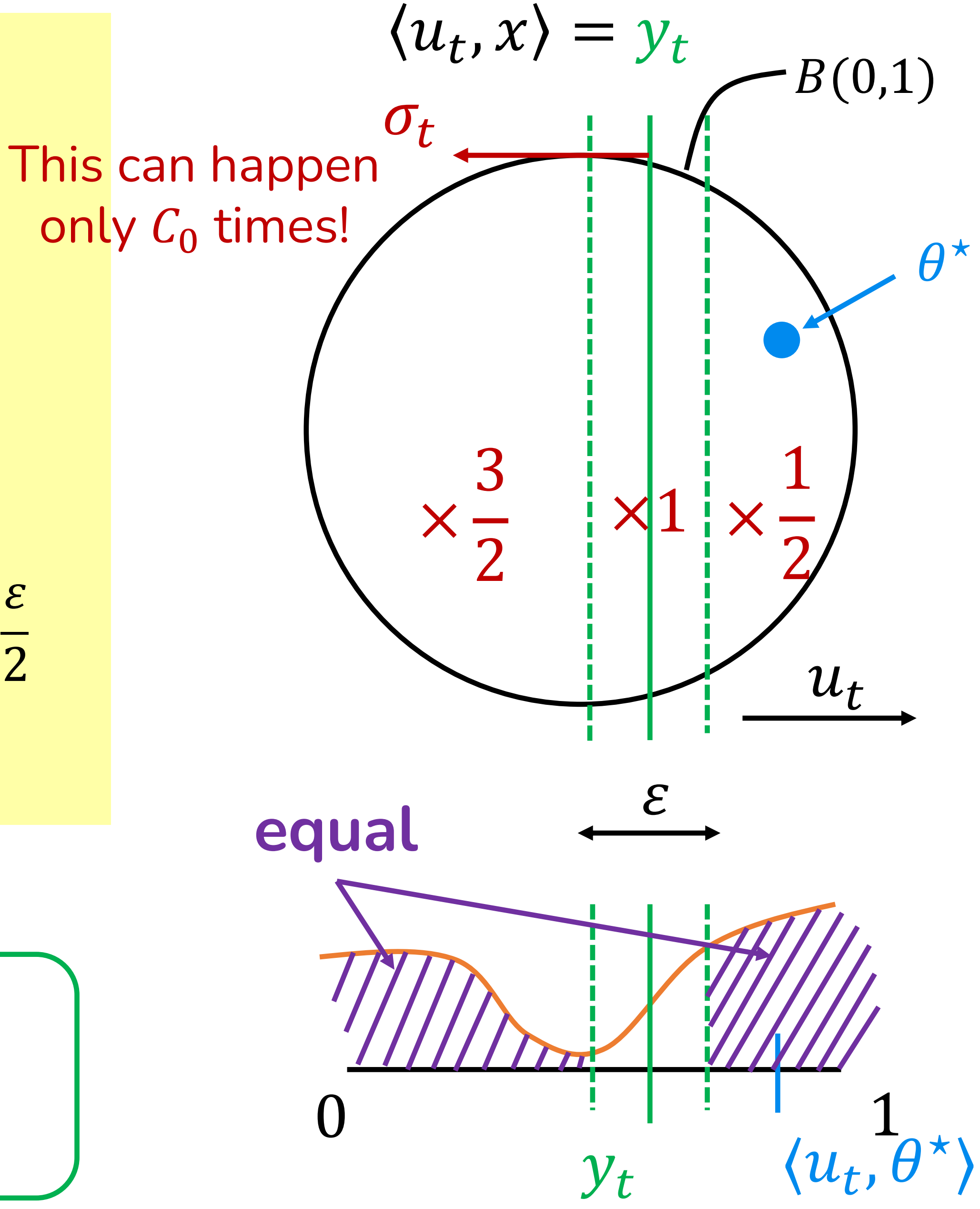
ε – Window Median Algorithm

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

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

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

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



Main Result

- ε – ball loss: $Regret = O(C_0 + d \log 1/\varepsilon)$

Algorithm for ε – Ball Loss

$\boldsymbol{\varepsilon}$ – Window Median Algorithm

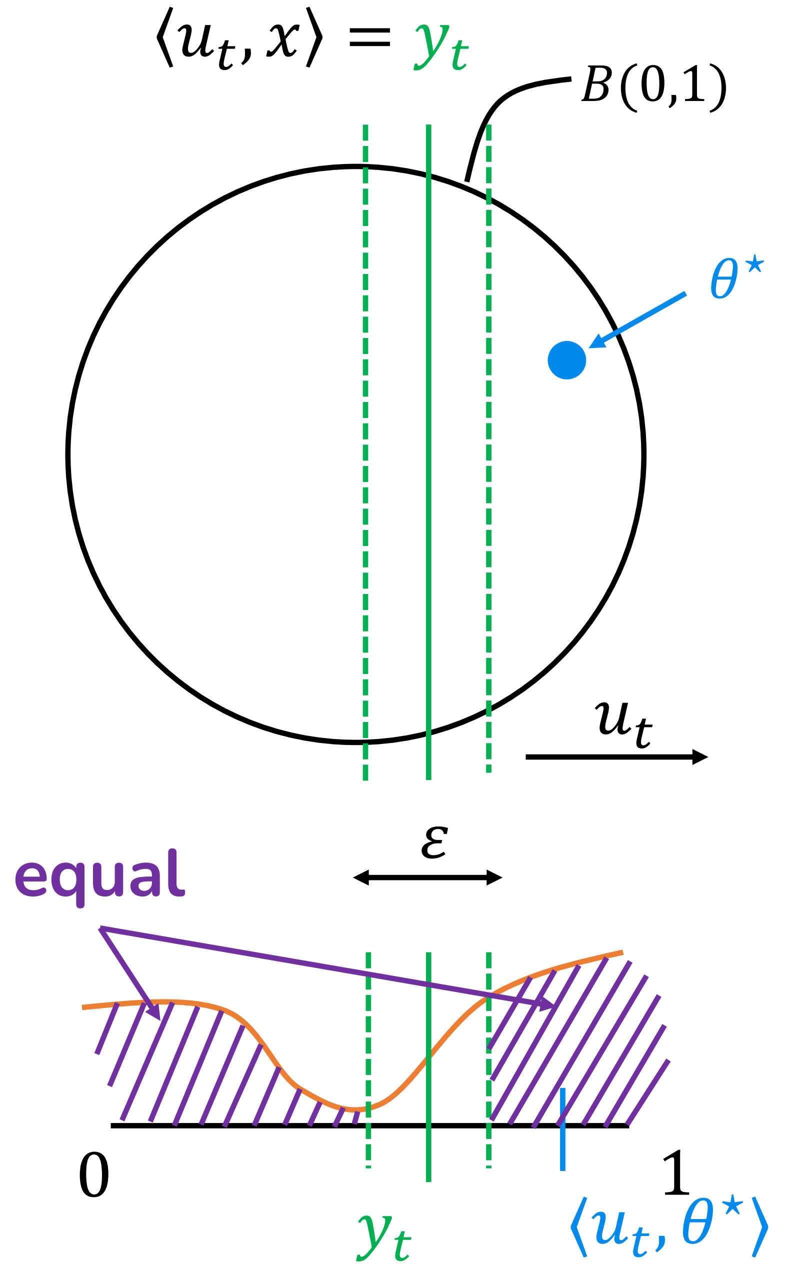
Initialize $f_1(x)$: uniform over $B(0,1)$.

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

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

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

Proof Idea



Algorithm for ε − Ball Loss

ε − Window Median Algorithm

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

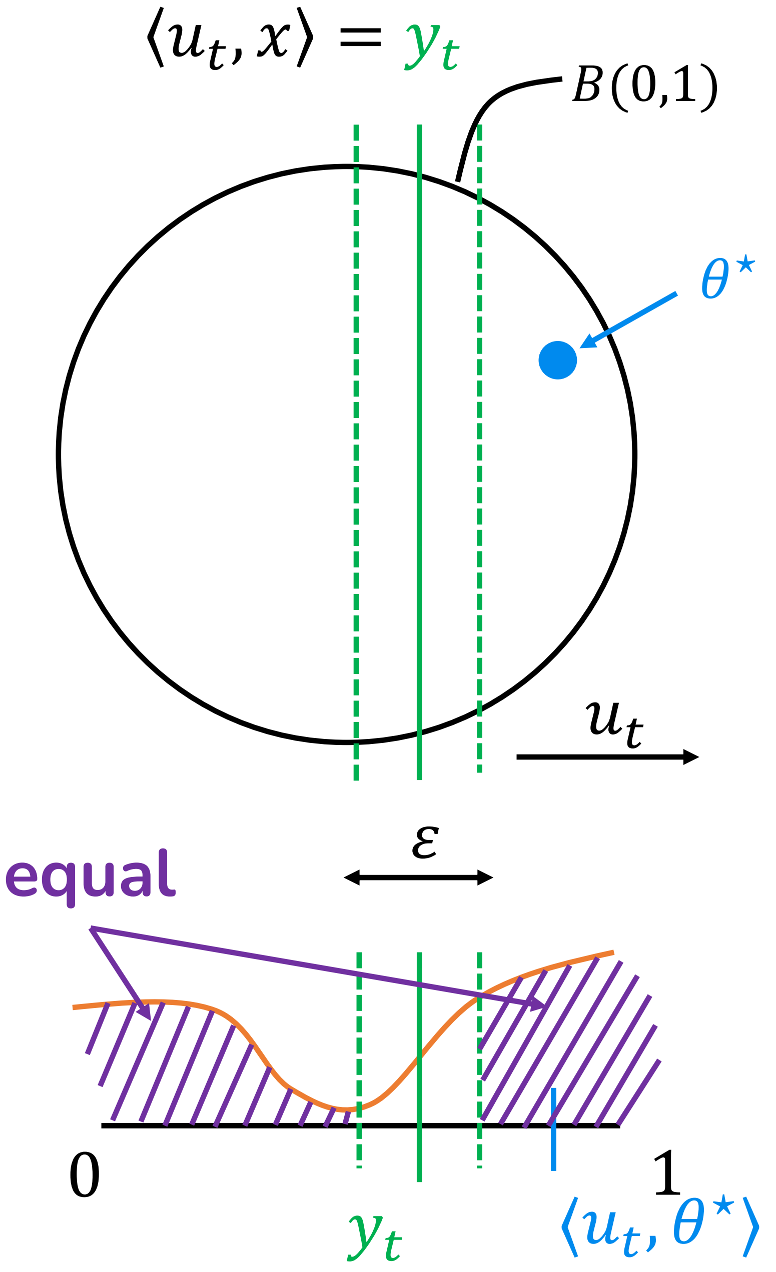
For rounds $t = 1, \dots, T$:

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

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

Proof Idea

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



Algorithm for ε – Ball Loss

ε – Window Median Algorithm

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

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

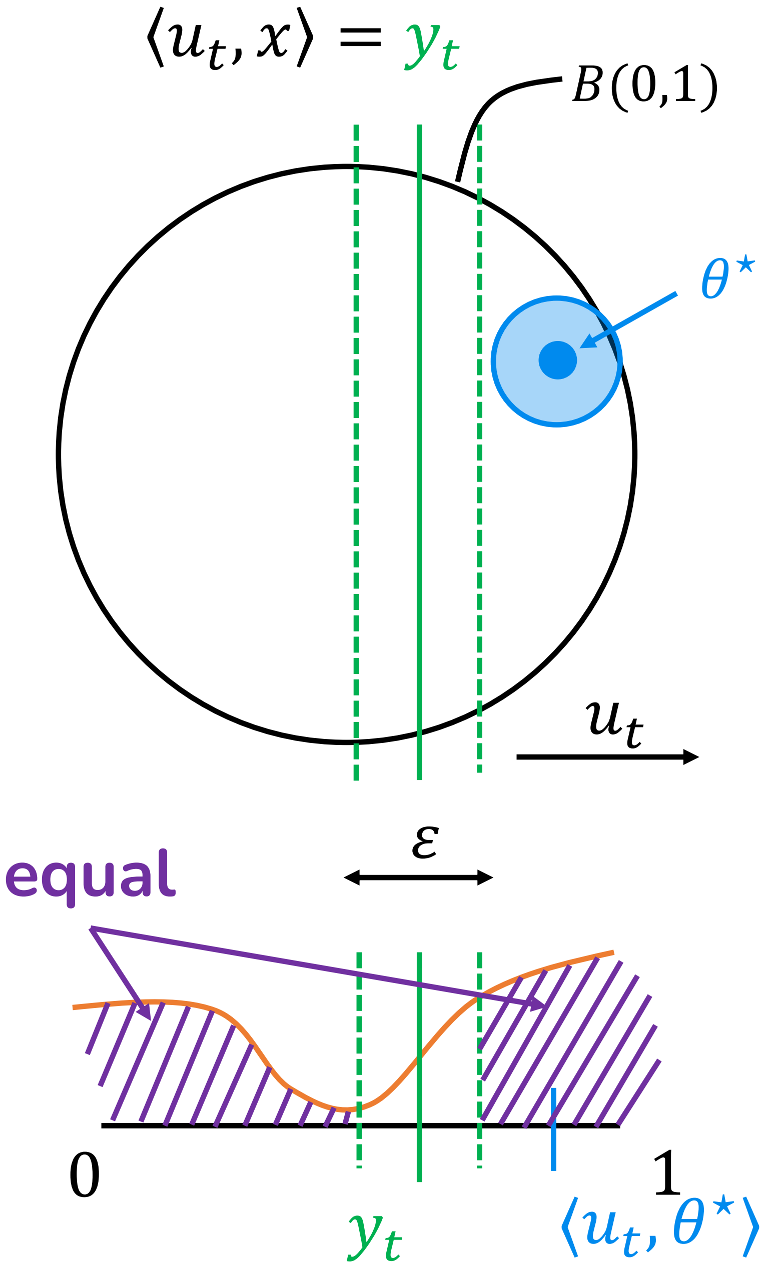
- Observe u_t and query $y_t = \varepsilon - window - median(f_t)$
- Update density:

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

Proof Idea

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

2. Potential $\Phi_t = \int_{B(\theta^\star, \varepsilon/2)} f_t(x)dx$:



Algorithm for ε – Ball Loss

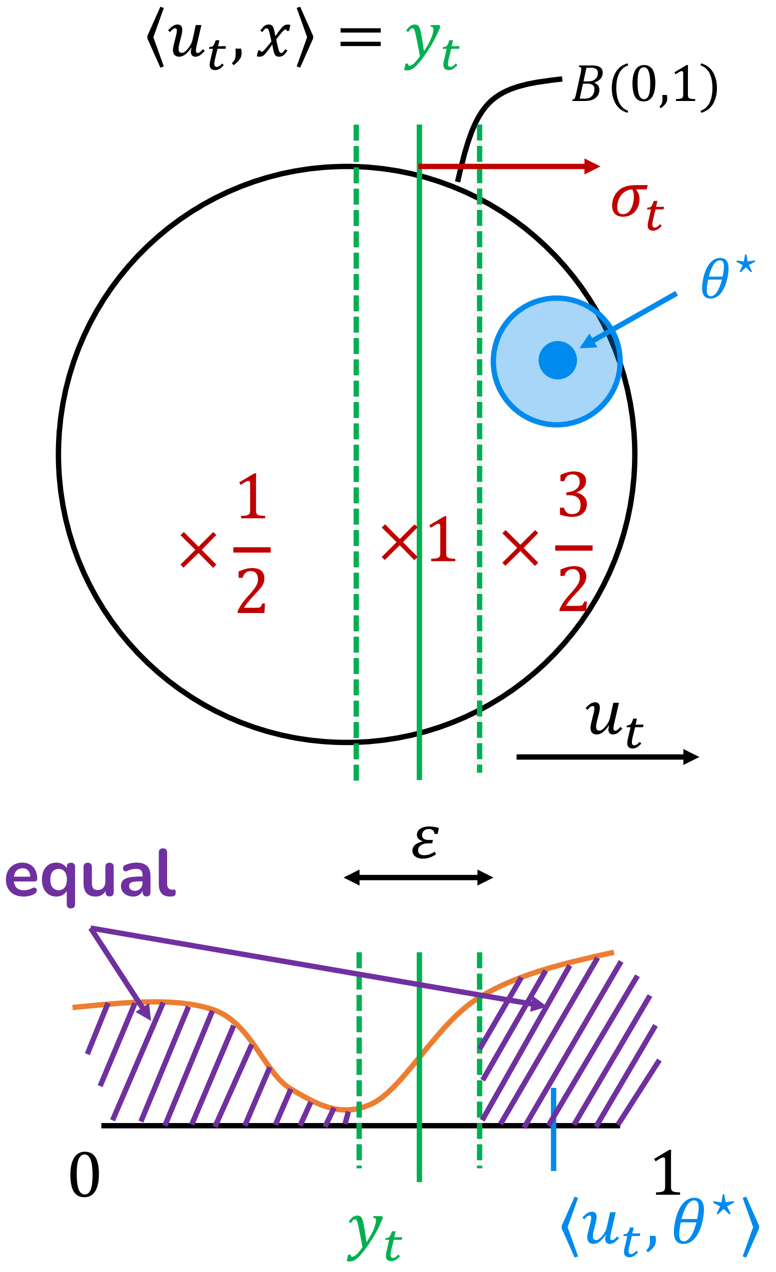
ε – Window Median Algorithm

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

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

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

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



Proof Idea

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

2. Potential $\Phi_t = \int_{B(\theta^\star, \varepsilon/2)} f_t(x)dx$:
   - **(weakly) increases** in uncorrupted rounds

Algorithm for ε – Ball Loss

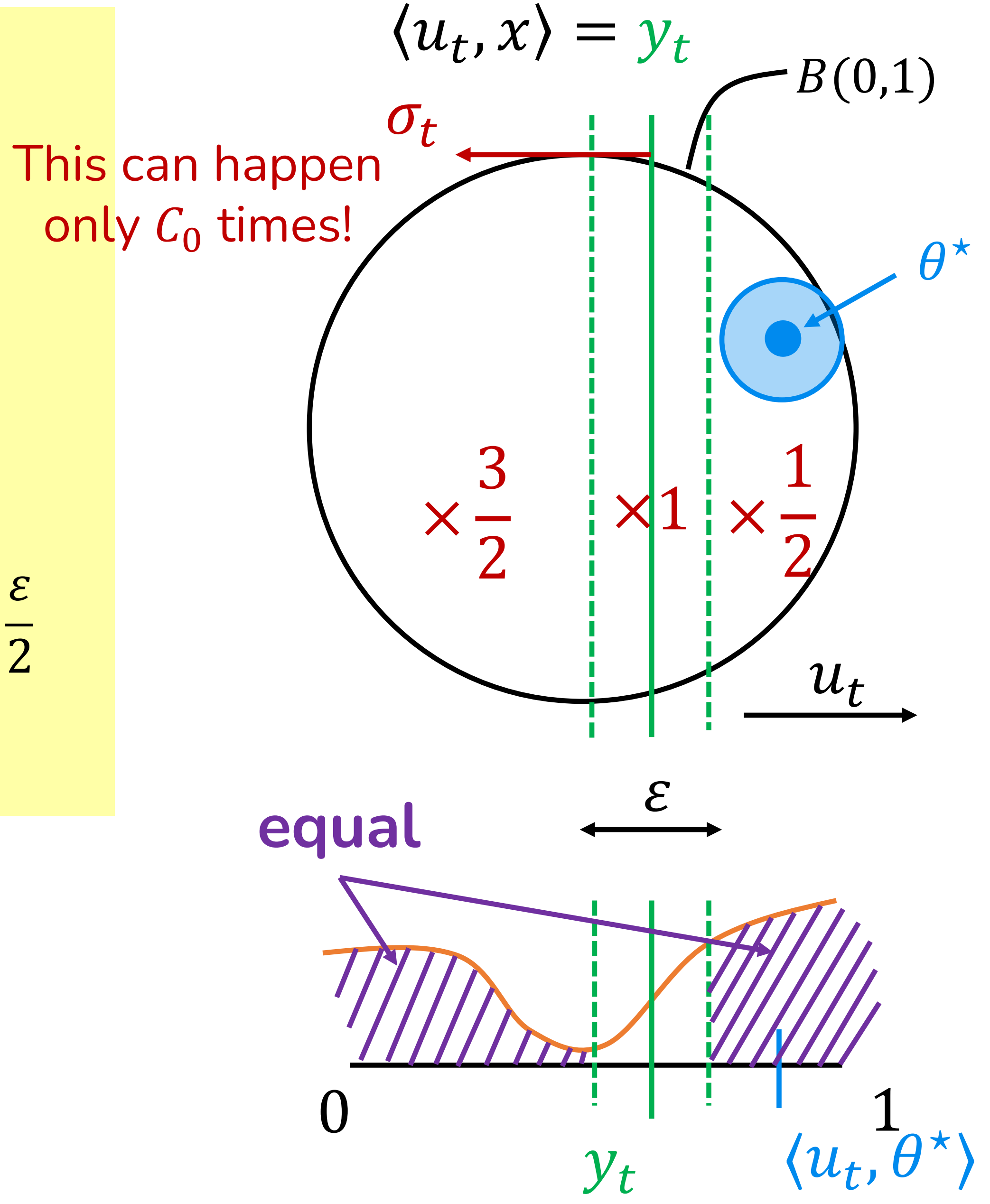
ε – Window Median Algorithm

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

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

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

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



Proof Idea

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

2. Potential $\Phi_t = \int_{B(\theta^\star, \varepsilon/2)} f_t(x)dx$:

- **(weakly) increases** in uncorrupted rounds
- **decreases by $1/2$** in corrupted ones (**C_0 in total**)

Algorithm for ε – Ball Loss

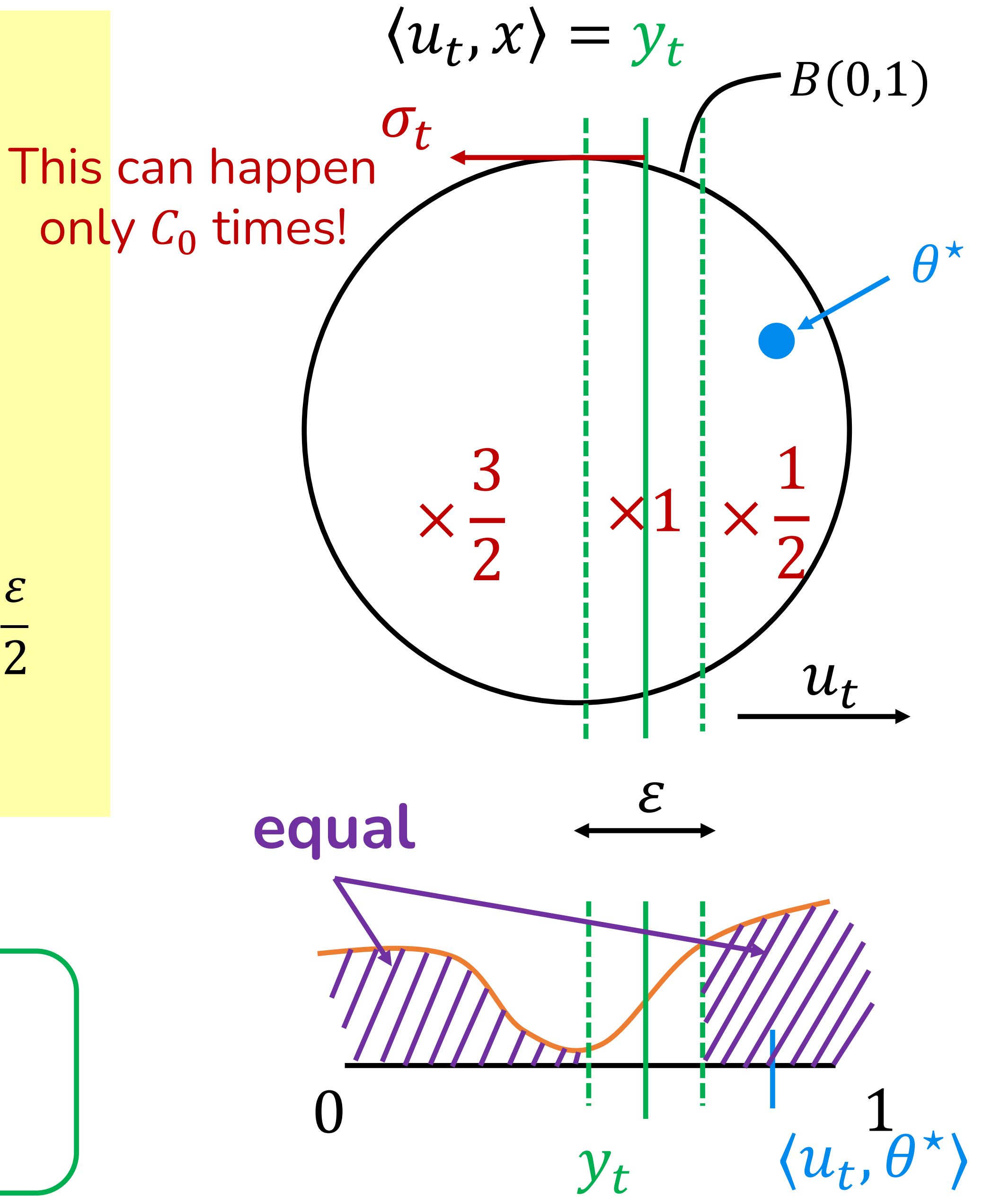
ε – Window Median Algorithm

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:

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



Main Result

- ε –ball loss: $Regret = O(C_0 + d \log 1/\varepsilon)$

Algorithm for ε – Ball Loss

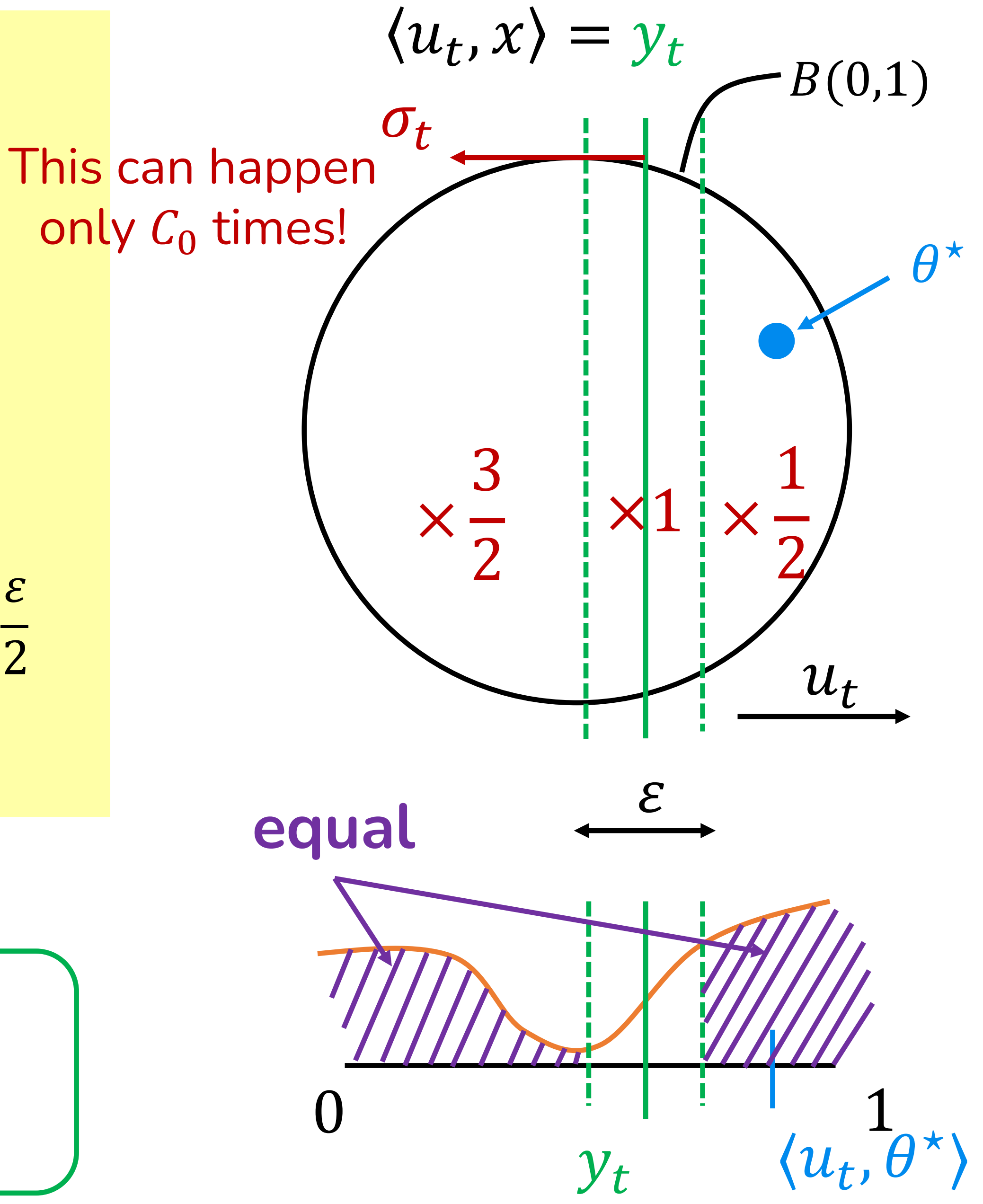
ε – Window Median Algorithm

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

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

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

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



Main Result

- ε –ball loss: $Regret = O(C_0 + d \log 1/\varepsilon)$
- symmetric loss: $Regret = O(C_0 + d \log T)$

Algorithm for ε − Ball Loss

ε − Window Median Algorithm

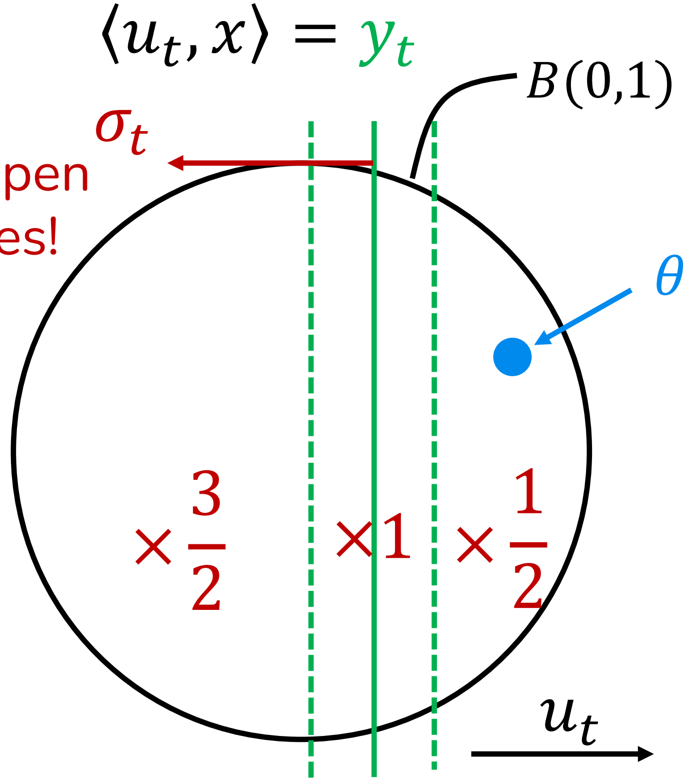
Initialize $f_1(x)$: uniform over $B(0,1)$.

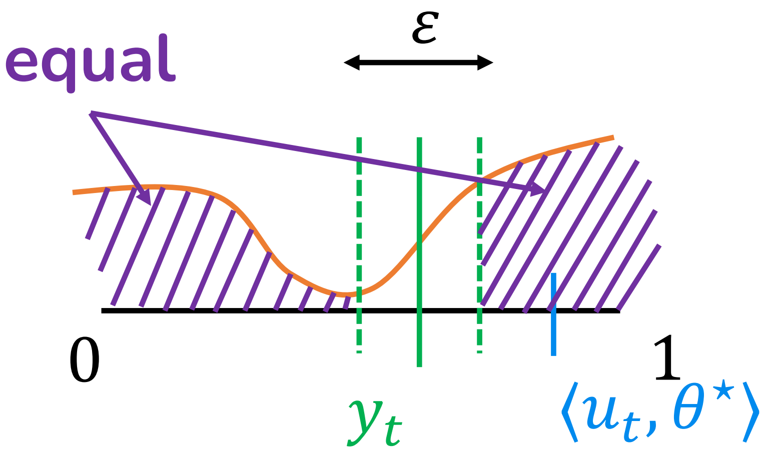
For rounds $t = 1, \ldots, T$:

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

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

This can happen only C_0 times!





Runtime ≈ $O(T^d\, poly(d,T))$

Main Result

- ε −ball loss: $Regret = O(C_0 + d \log 1/\varepsilon)$
- symmetric loss: $Regret = O(C_0 + d \log T)$

Efficient Algorithm for Symmetric Loss

Main Result

Polytime algorithm with $Regret = O(C_1 + d \log T)$ for symmetric loss,

where $C_1 = \sum_t |z_t|$.

Efficient Algorithm for Symmetric Loss

Main Result

Polytime algorithm with $Regret = O(C_1 + d \log T)$ 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 $Regret = O(C_1 + d \log T)$ for symmetric loss,

where $C_1 = \sum_t |z_t|$.

$C_1 < C_0 = \sum_t 1\{z_t \neq 0\}$

Idea

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

Efficient Algorithm for Symmetric Loss

Main Result

Polytime algorithm with $Regret = O(C_1 + d \log T)$ for symmetric loss,

where $C_1 = \sum_t |z_t|$.

$C_1 < C_0 = \sum_t 1\{z_t \neq 0\}$

Idea

efficient sampling
([Applegate & Kannan, STOC91])

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

Main Result

Corruption-robust contextual search algorithms with rates:

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

Main Result

Corruption-robust contextual search algorithms with rates:

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

Open Questions

Main Result

Corruption-robust contextual search algorithms with rates:

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

Open Questions

1. Variant of **distribution-based algorithms** for **pricing loss**.

Main Result

Corruption-robust contextual search algorithms with rates:

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

Open Questions

1. Variant of **distribution-based algorithms** for **pricing loss**.

2. Algorithms with $Regret = O(C_1 + d \log d)$ for **symmetric loss**.

Main Result

Corruption-robust contextual search algorithms with rates:

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

Open Questions

1. Variant of **distribution-based algorithms** for **pricing loss**.

2. Algorithms with $Regret = O(C_1 + d \log d)$ for **symmetric loss**.

3. **Polytime** algorithm for **ε –ball loss**.

Main Result

Corruption-robust contextual search algorithms with rates:

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

Open Questions

1. Variant of **distribution-based algorithms** for **pricing loss**.

2. Algorithms with $Regret = O(C_1 + d \log d)$ for **symmetric loss**.

3. **Polytime** algorithm for **ε –ball loss**.