Improved Regret Guarantees for OCO in Bandit Setting

Ankan Saha [1] Ambuj Tewari [2]

AISTATS 2011

[1]University of Chicago
[2]University of Texas at Austin

Introduction and Motivation

- Sequential decision making: important question in machine learning, Economics, Operations Research and related fields.
- Modeled as a sequential game between learner and adversary.

Setting

- At every time step t:

Setting

- At every time step t:
- Player plays a point $\mathbf{x}_t \in \mathcal{K} \subseteq \mathbb{R}^d$, $\mathcal{K}$ is convex and compact.

Setting

- At every time step t:
- Player plays a point $\mathbf{x}_t \in \mathcal{K} \subseteq \mathbb{R}^d$, $\mathcal{K}$ is convex and compact.
- Adversary responds with a function $f_t \in \mathcal{F}$.
- Player suffers a loss $f_t(\mathbf{x}_t)$.

Setting

- At every time step t:
- Player plays a point $\mathbf{x}_t \in \mathcal{K} \subseteq \mathbb{R}^d$, $\mathcal{K}$ is convex and compact.
- Adversary responds with a function $f_t \in \mathcal{F}$.
- Player suffers a loss $f_t(\mathbf{x}_t)$.
- *Online* Convex Optimization: f_t are convex functions.

Setting

- At every time step t:
- Player plays a point $\mathbf{x}_t \in \mathcal{K} \subseteq \mathbb{R}^d$, $\mathcal{K}$ is convex and compact.
- Adversary responds with a function $f_t \in \mathcal{F}$.
- Player suffers a loss $f_t(\mathbf{x}_t)$.
- *Online* Convex Optimization: f_t are convex functions.
- Full information vs. Bandit Setting.

- Goal: To minimize the **Regret** *i.e.* the player's performance with respect to the best performance in hindsight.

$$R_T = \sum_{i=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}^\star \in \mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{x}^\star)$$

- Goal: To minimize the **Regret** *i.e.* the player's performance with respect to the best performance in hindsight.

$$R_T = \sum_{i=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}^\star \in \mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{x}^\star)$$

- Full information Setting: Zinkevich's algorithm.

$$\mathbf{x}_{t+1} = \mathbf{Proj}_{\mathcal{K}} \left\{ \mathbf{x}_t - \eta_t \nabla f_t(\mathbf{x}_t) \right\}$$

- Goal: To minimize the **Regret** *i.e.* the player's performance with respect to the best performance in hindsight.

$$R_T = \sum_{i=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}^\star \in \mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{x}^\star)$$

- Full information Setting: Zinkevich's algorithm.

$$\mathbf{x}_{t+1} = \mathbf{Proj}_{\mathcal{K}} \left\{ \mathbf{x}_t - \eta_t \nabla f_t(\mathbf{x}_t) \right\}$$

- Projected gradient descent strategy incurs $O(\sqrt{T})$ regret.
- Zinkevich's algorithm just requires information about the gradients of f_t at every time step.

Zinkevich's Algorithm

- Key lies in the fact that we minimize the regret and not the actual loss.

Theorem

If the updates are given by $\mathbf{x}_{t+1} = \mathbf{Proj}_{\mathcal{K}}\left(\mathbf{x}_t - \eta_t \nabla f_t(\mathbf{x}_t)\right)$, *choosing* $\eta_t = t^{-1/2}$ *gives the following bound on regret after* T *steps*

$$R_T \leq \frac{diam(\mathcal{K})^2 \sqrt{T}}{2} + (\sqrt{T} - \frac{1}{2}) \left\|\nabla f\right\|^2$$

where $\left\|\nabla f\right\| = \max_{\mathbf{x}\in\mathcal{K}, t\in[T]} \left\|\nabla f_t(\mathbf{x})\right\|$

Zinkevich's Algorithm

- Key lies in the fact that we minimize the regret and not the actual loss.

Theorem

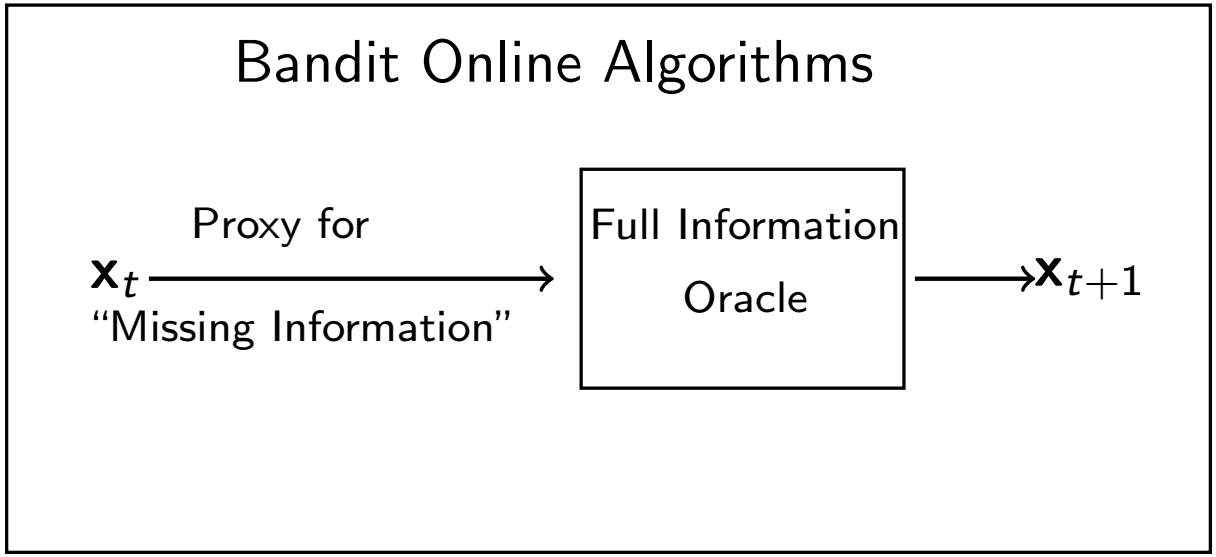
If the updates are given by $\mathbf{x}_{t+1} = \mathbf{Proj}_{\mathcal{K}}\left(\mathbf{x}_t - \eta_t \nabla f_t(\mathbf{x}_t)\right)$, *choosing* $\eta_t = t^{-1/2}$ *gives the following bound on regret after* T *steps*

$$R_T \leq \frac{diam(\mathcal{K})^2 \sqrt{T}}{2} + (\sqrt{T} - \frac{1}{2}) \left\|\nabla f\right\|^2$$

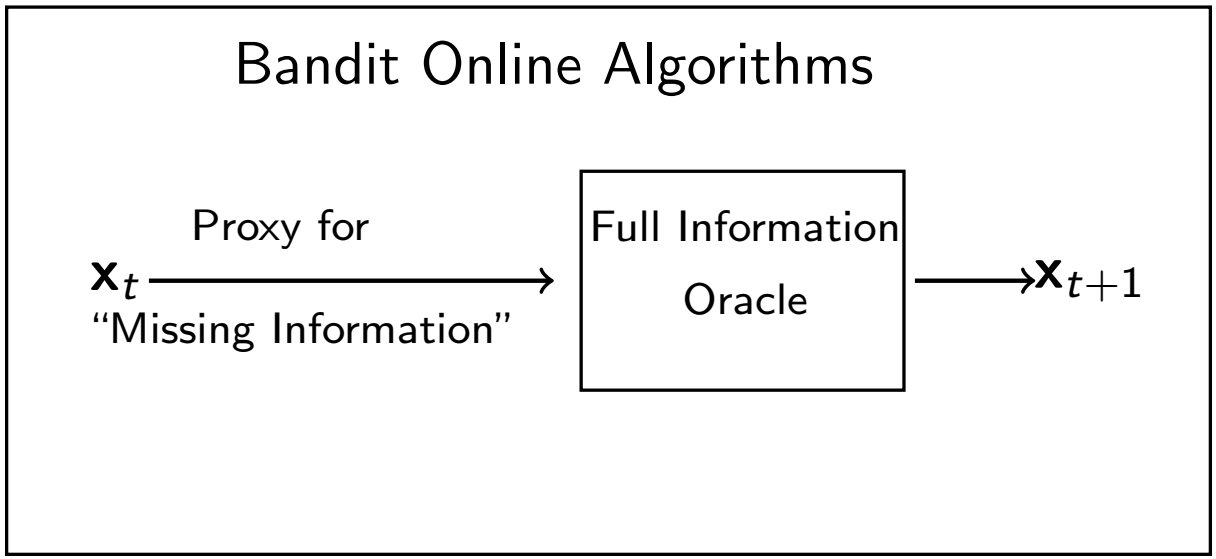
where $\left\|\nabla f\right\| = \max_{\mathbf{x}\in\mathcal{K}, t\in[T]} \left\|\nabla f_t(\mathbf{x})\right\|$

- Convergence rate depends on $\left\|\nabla f\right\|$.

Bandit Online Optimization

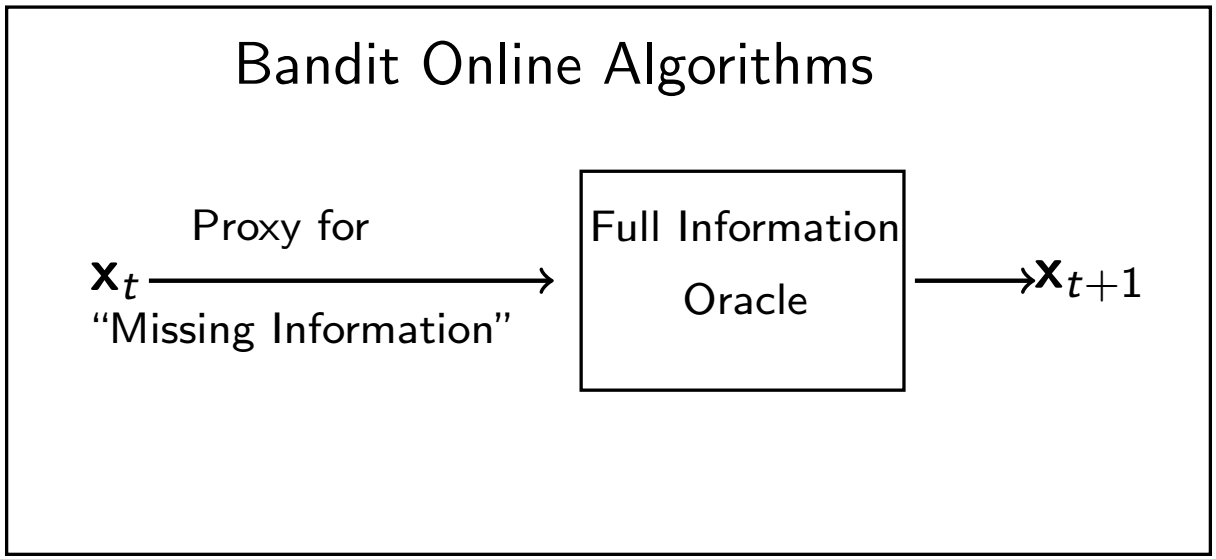


Bandit Online Optimization



- Missing information $\Longrightarrow$ Gradient of the function.

Bandit Online Optimization



- Missing information $\Longrightarrow$ Gradient of the function.
- How to evaluate the gradient from a single point evaluation?

Unbiased gradient estimation

- [FKM05] provides such a scheme.
- Introduce randomness.

$$\hat{f}(\mathbf{x}) = \mathbb{E}_{\mathbf{v} \in \mathbb{B}^d} \left[f(\mathbf{x} + \delta \mathbf{v}) \right]$$

Unbiased gradient estimation

- [FKM05] provides such a scheme.
- Introduce randomness.

$$\hat{f}(\mathbf{x}) = \mathbb{E}_{\mathbf{v} \in \mathbb{B}^d} \left[f(\mathbf{x} + \delta \mathbf{v}) \right]$$

- Then

$$\nabla \hat{f}(\mathbf{x}) = \mathbb{E}_{\mathbf{u} \in \mathbb{S}^d} \left[\frac{d}{\delta} f(\mathbf{x} + \delta \mathbf{u}).\mathbf{u} \right]$$

Problems with existing approach

- Key trick in [FKM05] is to choose the size of the ball for evaluating the unbiased gradient estimate.

Problems with existing approach

- Key trick in [FKM05] is to choose the size of the ball for evaluating the unbiased gradient estimate.
- Too small δ blows up the $\|\nabla f\|$, depending on the location of $\mathbf{x}_t$ and weakens the bounds.

Problems with existing approach

- Key trick in [FKM05] is to choose the size of the ball for evaluating the unbiased gradient estimate.
- Too small δ blows up the $\|\nabla f\|$, depending on the location of $\mathbf{x}_t$ and weakens the bounds.
- Too large $\delta \implies$ Gradient estimates are not very accurate.

Problems with existing approach

- Key trick in [FKM05] is to choose the size of the ball for evaluating the unbiased gradient estimate.
- Too small δ blows up the $\|\nabla f\|$, depending on the location of $\mathbf{x}_t$ and weakens the bounds.
- Too large $\delta \implies$ Gradient estimates are not very accurate.
- Trading off the ball size along with Zinkevich's scheme gives $O(T^{3/4})$ regret in the OCO bandit setting.

Convex Analysis Basics

Convex Analysis Basics

- Convex functions: Lower Bounded by linear approximation.

$$f(\mathbf{x}) \geq f(\mathbf{y}) + \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle$$

Convex Analysis Basics

- Convex functions: Lower Bounded by linear approximation.

$$f(\mathbf{x}) \geq f(\mathbf{y}) + \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle$$

- Strongly Convex functions: Lower Bounded by a quadratic. Eg. 2-norm squared.

$$f(\mathbf{x}) \geq f(\mathbf{y}) + \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{\sigma}{2} \|\mathbf{x} - \mathbf{y}\|^2$$

Convex Analysis Basics

- Convex functions: Lower Bounded by linear approximation.

$$f(\mathbf{x}) \geq f(\mathbf{y}) + \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle$$

- Strongly Convex functions: Lower Bounded by a quadratic. Eg. 2-norm squared.

$$f(\mathbf{x}) \geq f(\mathbf{y}) + \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{\sigma}{2} \|\mathbf{x} - \mathbf{y}\|^2$$

- *l.c.g* functions: Upper Bounded by a quadratic.

$$f(\mathbf{x}) \leq f(\mathbf{y}) + \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{L}{2} \|\mathbf{x} - \mathbf{y}\|^2$$

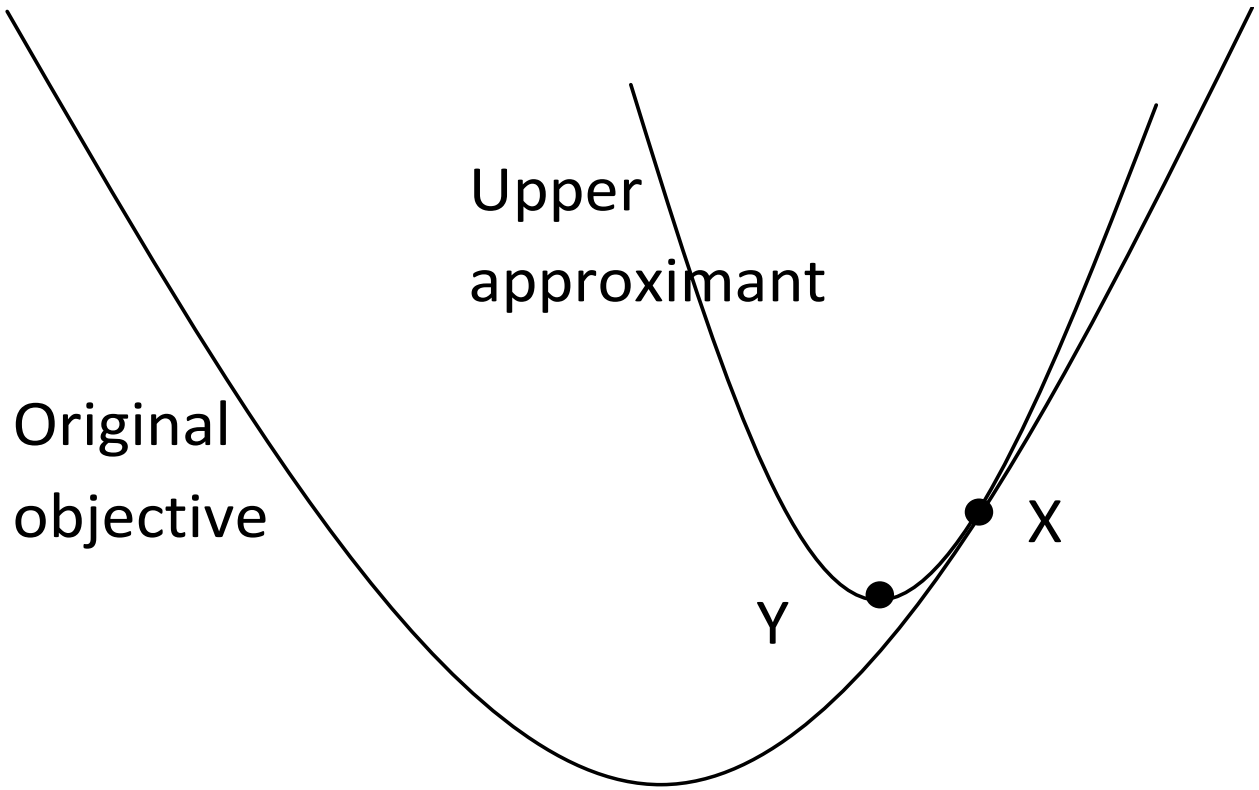
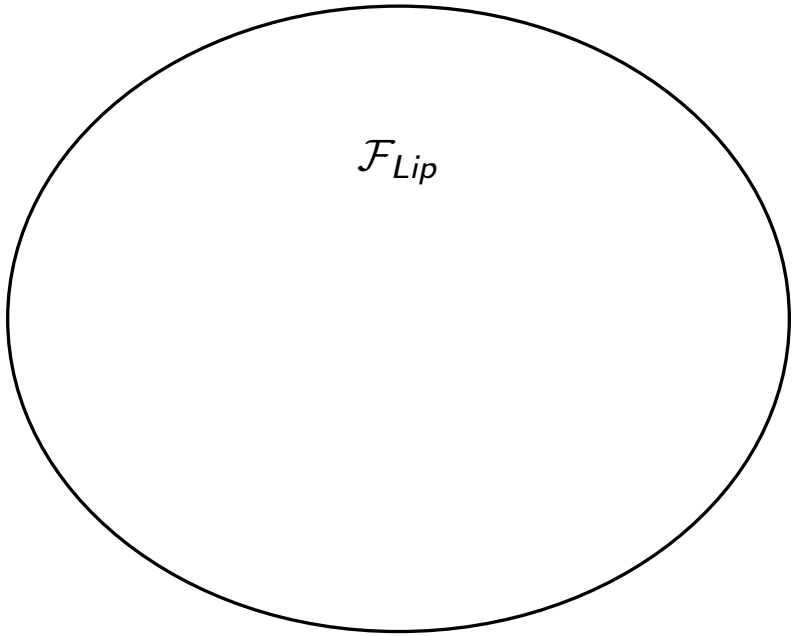


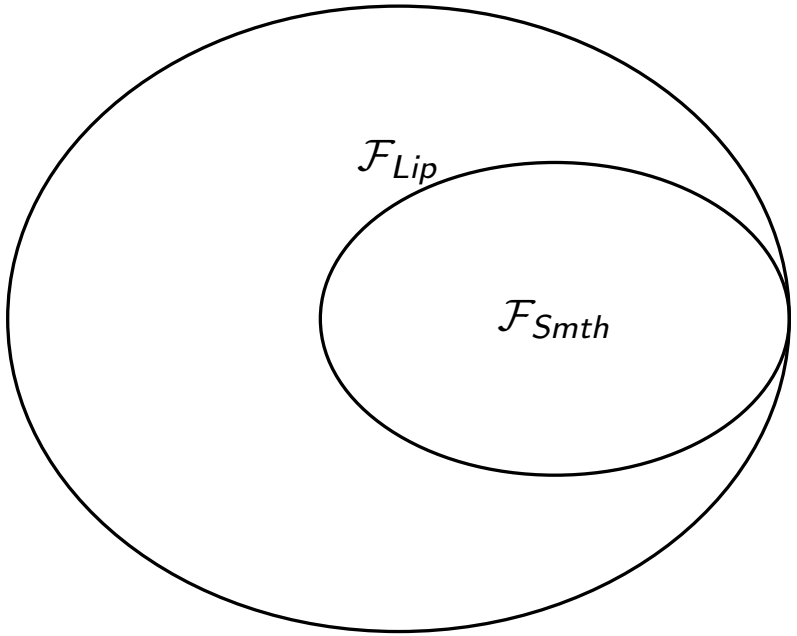
Figure: *l.c.g* functions

Results for Subclasses of Convex functions(Full Information Setting)



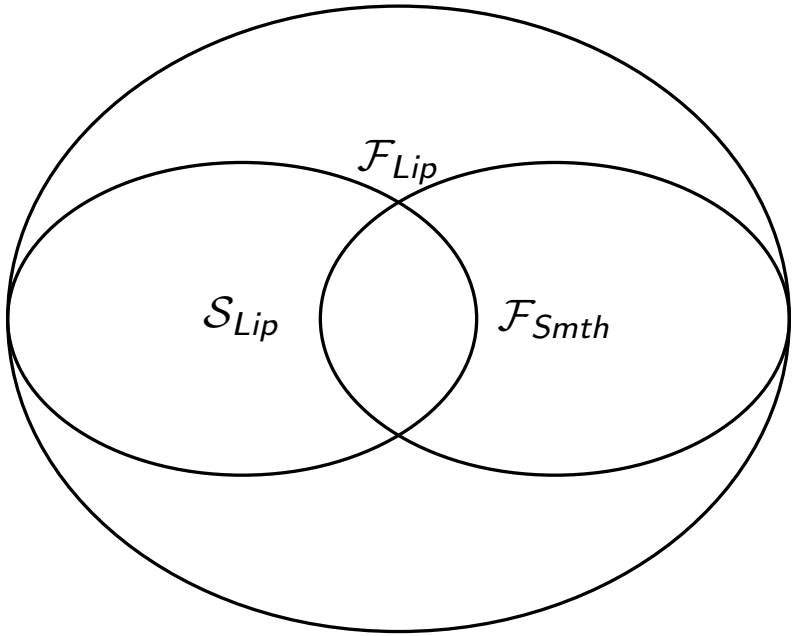
Function Class	Best Rates on Regret
Lipschitz ($\mathcal{F}_{Lip}$)	$\Theta(T^{1/2})$ [Zin03]

Results for Subclasses of Convex functions(Full Information Setting)



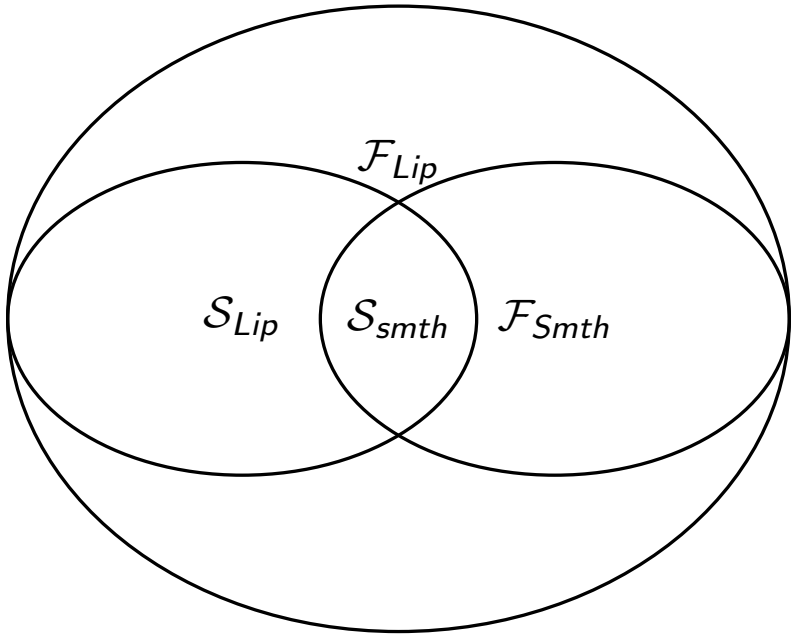
Function Class	Best Rates on Regret
Lipschitz ($\mathcal{F}_{Lip}$)	$\Theta(T^{1/2})$ [Zin03]
Smooth ($\mathcal{F}_{Smth}$)	$\Theta(T^{1/2})$ [Zin03]

Results for Subclasses of Convex functions(Full Information Setting)



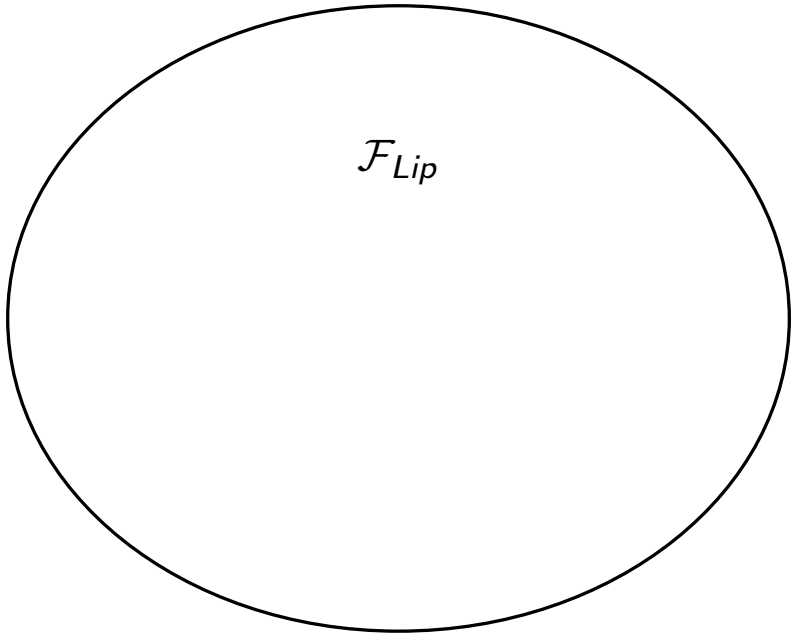
Function Class	Best Rates on Regret
Lipschitz ($\mathcal{F}_{Lip}$)	$\Theta(T^{1/2})$ [Zin03]
Smooth ($\mathcal{F}_{Smth}$)	$\Theta(T^{1/2})$ [Zin03]
Lipschitz and Strongly Convex ($\mathcal{S}_{Lip}$)	$\Theta(\log T)$ [HAK07]

Results for Subclasses of Convex functions(Full Information Setting)



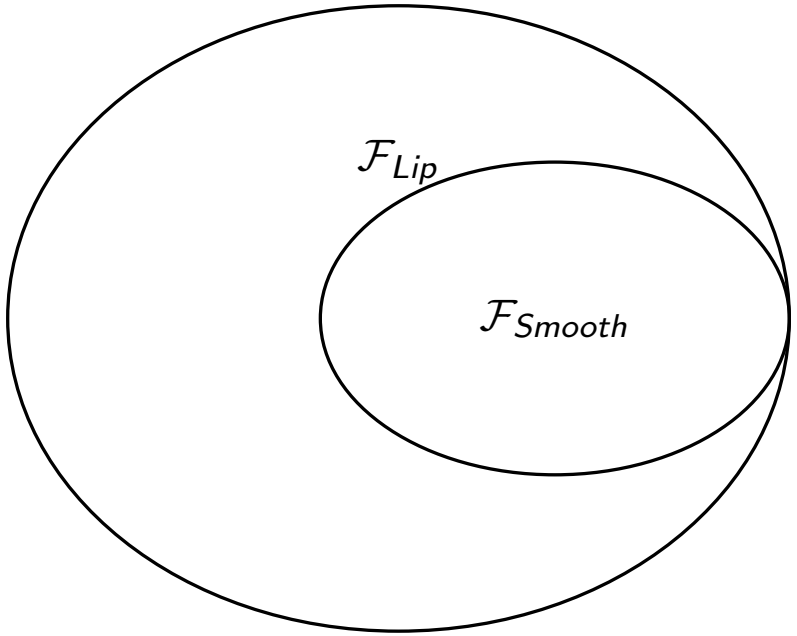
Function Class	Best Rates on Regret
Lipschitz ($\mathcal{F}_{Lip}$)	$\Theta(T^{1/2})$ [Zin03]
Smooth ($\mathcal{F}_{Smth}$)	$\Theta(T^{1/2})$ [Zin03]
Lipschitz and Strongly Convex ($\mathcal{S}_{Lip}$)	$\Theta(\log T)$ [HAK07]
Smooth and Strongly Convex ($\mathcal{S}_{Smth}$)	$\Theta(\log T)$ [HAK07]

Results for Bandit Setting



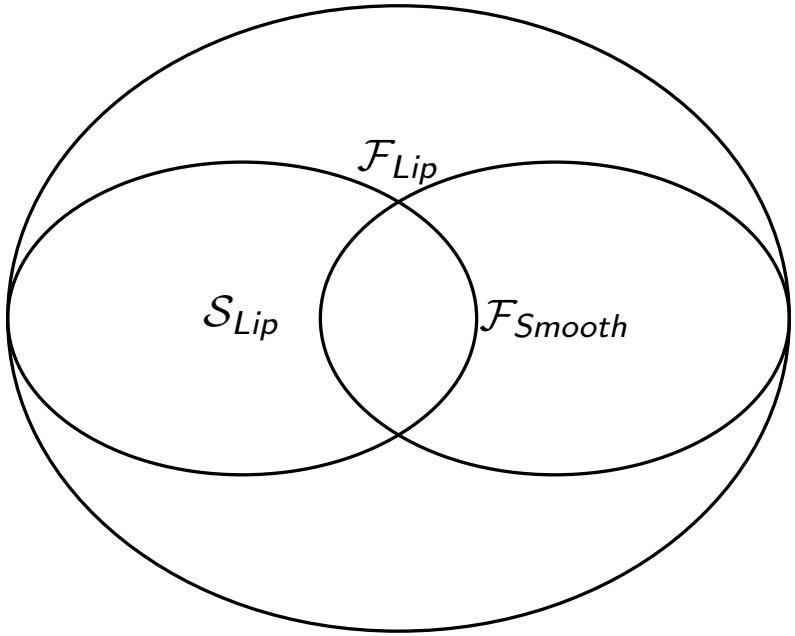
Function Class	Best Rates on Regret	Lower Bounds
Lipschitz ($\mathcal{F}_{Lip}$)	$O(T^{3/4})$ [FKM05]	$\Omega(T^{1/2})$

Results for Bandit Setting



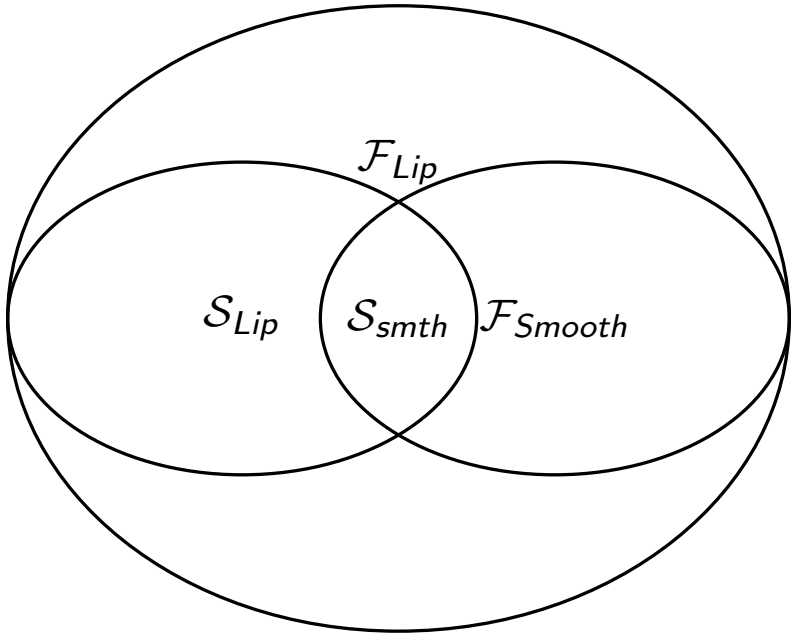
Function Class	Best Rates on Regret	Lower Bounds
Lipschitz ($\mathcal{F}_{Lip}$)	$O(T^{3/4})$ [FKM05]	$\Omega(T^{1/2})$
Smooth ($\mathcal{F}_{Smth}$)	$O^*(T^{2/3})$ **[This paper]**	$\Omega(T^{1/2})$

Results for Bandit Setting



Function Class	Best Rates on Regret	Lower Bounds
Lipschitz ($\mathcal{F}_{Lip}$)	$O(T^{3/4})$ [FKM05]	$\Omega(T^{1/2})$
Smooth ($\mathcal{F}_{Smth}$)	$O^*(T^{2/3})$ **[This paper]**	$\Omega(T^{1/2})$
Lipschitz and S.C. ($\mathcal{S}_{Lip}$)	$O^*(T^{2/3})$ [ADX10]	$\Omega(\log T)$

Results for Bandit Setting



Function Class	Best Rates on Regret	Lower Bounds
Lipschitz ($\mathcal{F}_{Lip}$)	$O(T^{3/4})$ [FKM05]	$\Omega(T^{1/2})$
Smooth ($\mathcal{F}_{Smth}$)	$O^*(T^{2/3})$ **[This paper]**	$\Omega(T^{1/2})$
Lipschitz and S.C. ($\mathcal{S}_{Lip}$)	$O^*(T^{2/3})$ [ADX10]	$\Omega(\log T)$
Smooth and S.C. ($\mathcal{S}_{Smth}$)	$O^*(T^{2/3})$ [ADX10]	$\Omega(\log T)$

New Approach

- Tighter bounds obtained by calculating $\|\nabla f\|$ with respect to a changing *local* norm, demonstrated for Bandit OLO [AHR08].

Definition

A function $R : \mathcal{K} \to \mathbb{R}$ is a self-concordant barrier if
a) $R \to \infty$ near the boundary of $\mathcal{K}$ and
b) R and $\nabla^2 R$ are Lipschitz continuous with respect to the local norm $\|\cdot\|_{R,\mathbf{w}}$, given by $\|f\|_{R,\mathbf{w}} = \sqrt{\langle f, \nabla^2 R(\mathbf{w}) f \rangle}$.

New Approach

- Tighter bounds obtained by calculating $\|\nabla f\|$ with respect to a changing *local* norm, demonstrated for Bandit OLO [AHR08].

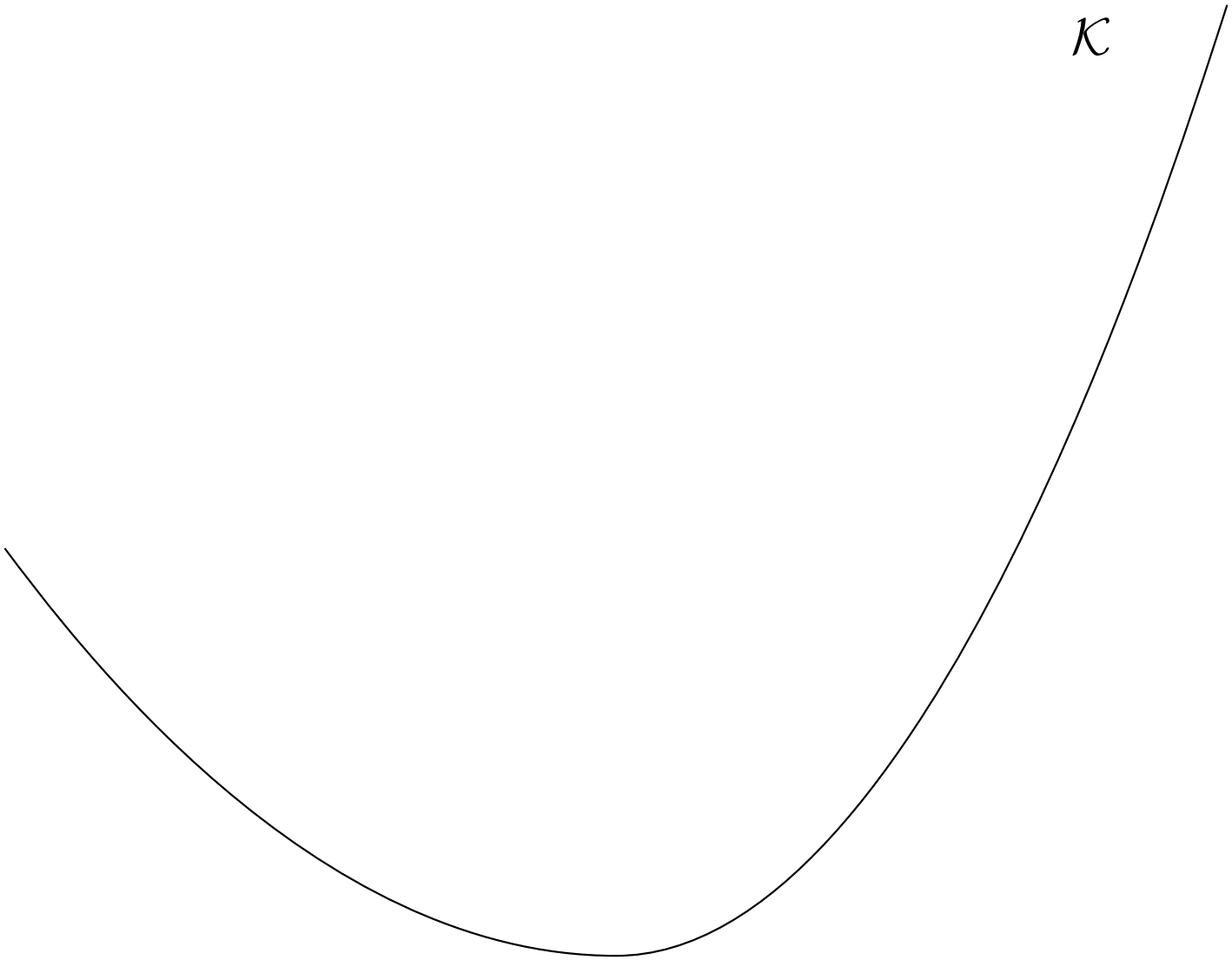
Definition

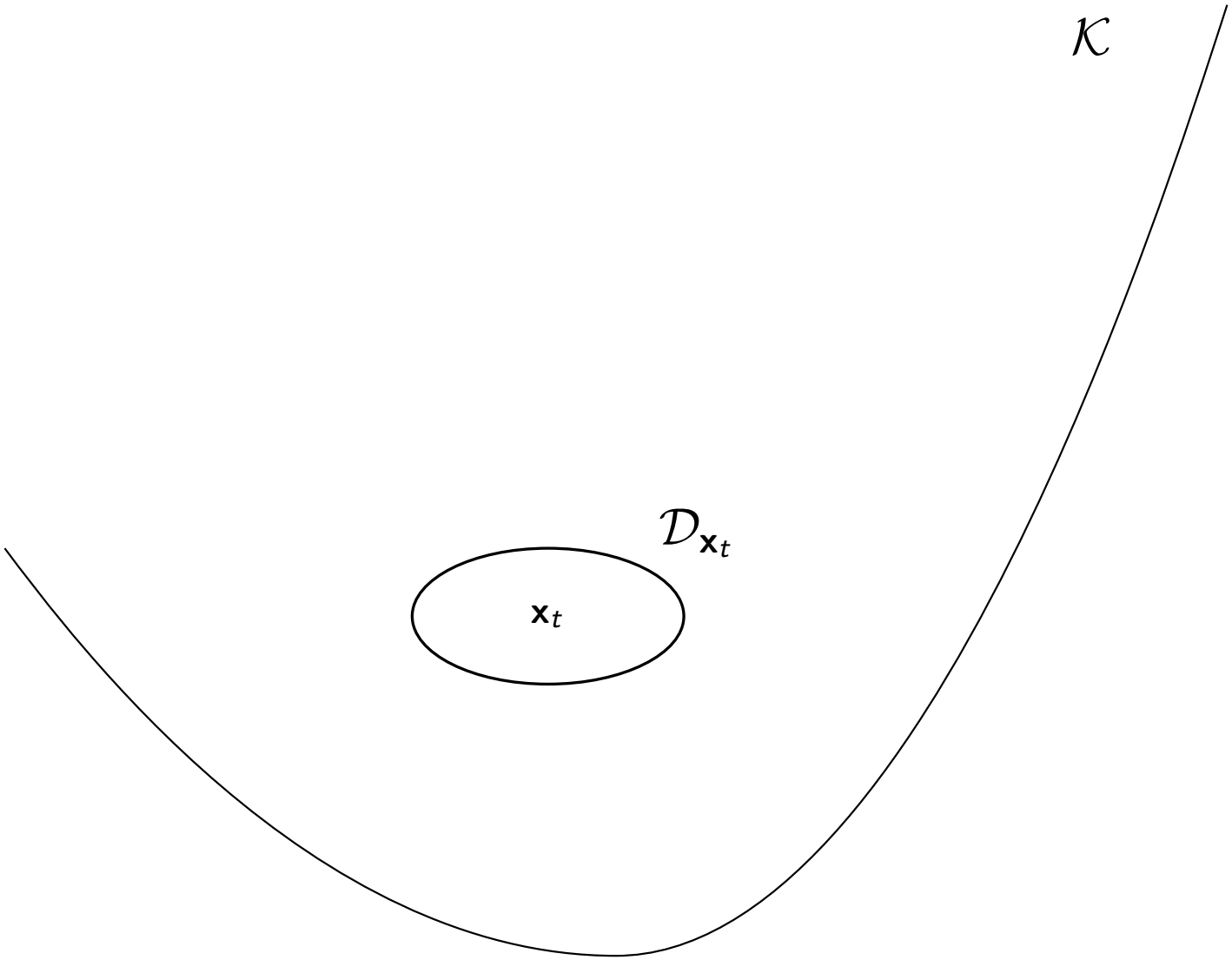
A function $R : \mathcal{K} \to \mathbb{R}$ is a self-concordant barrier if
a) $R \to \infty$ near the boundary of $\mathcal{K}$ and
b) R and $\nabla^2 R$ are Lipschitz continuous with respect to the local norm $\|\cdot\|_{R,\mathbf{w}}$, given by $\|f\|_{R,\mathbf{w}} = \sqrt{\langle f, \nabla^2 R(\mathbf{w}) f\rangle}$.

- Given a self concordant barrier R, for every $\mathbf{w} \in \mathcal{K}$, the Dikin ellipsoid centered at $\mathbf{w}$

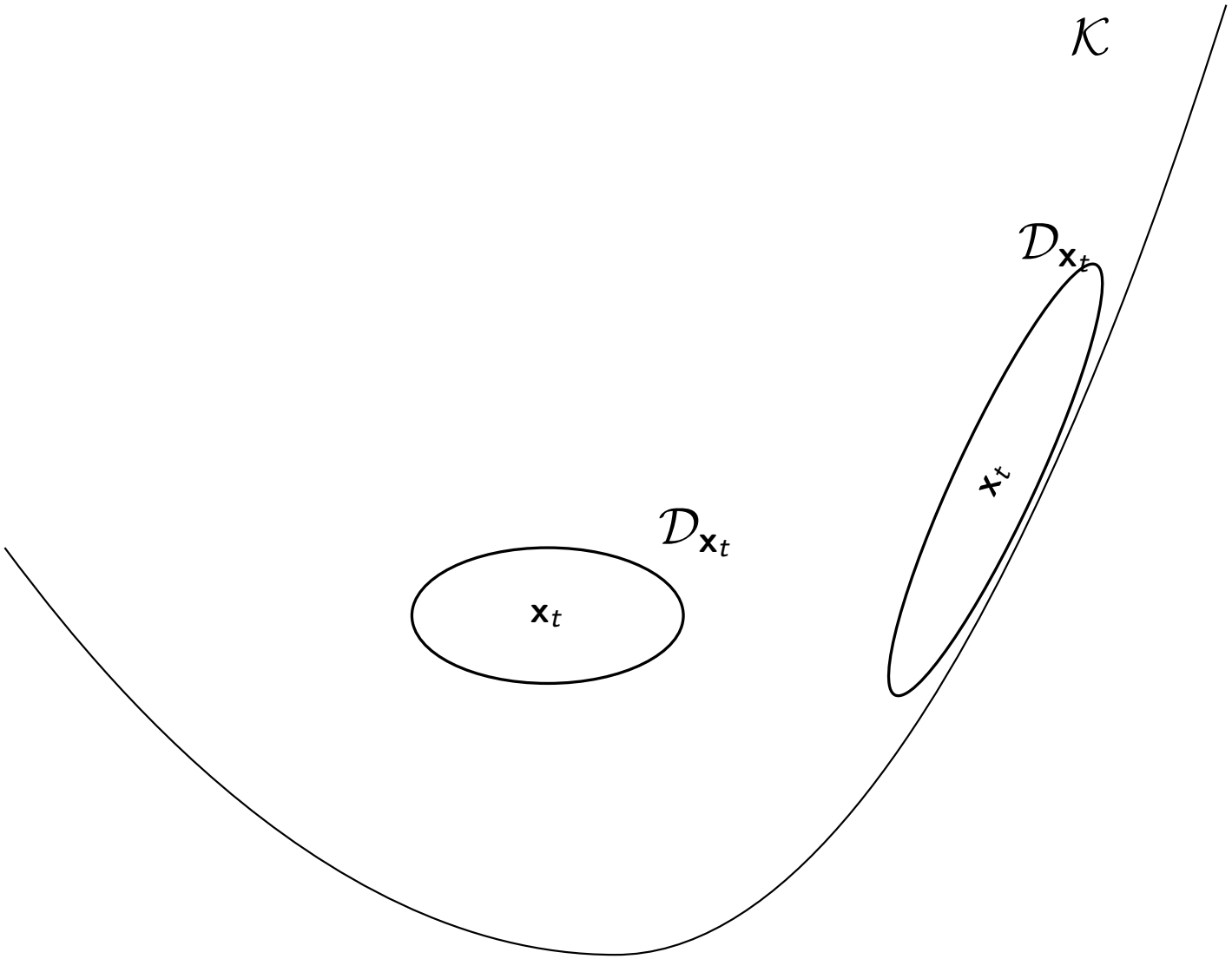
$$\mathcal{D}_{\mathbf{w}} = \left\{\mathbf{w}' : \left\|\mathbf{w} - \mathbf{w}'\right\|_{R,\mathbf{w}} \leq 1\right\}$$

is always contained in $\mathcal{K}$.

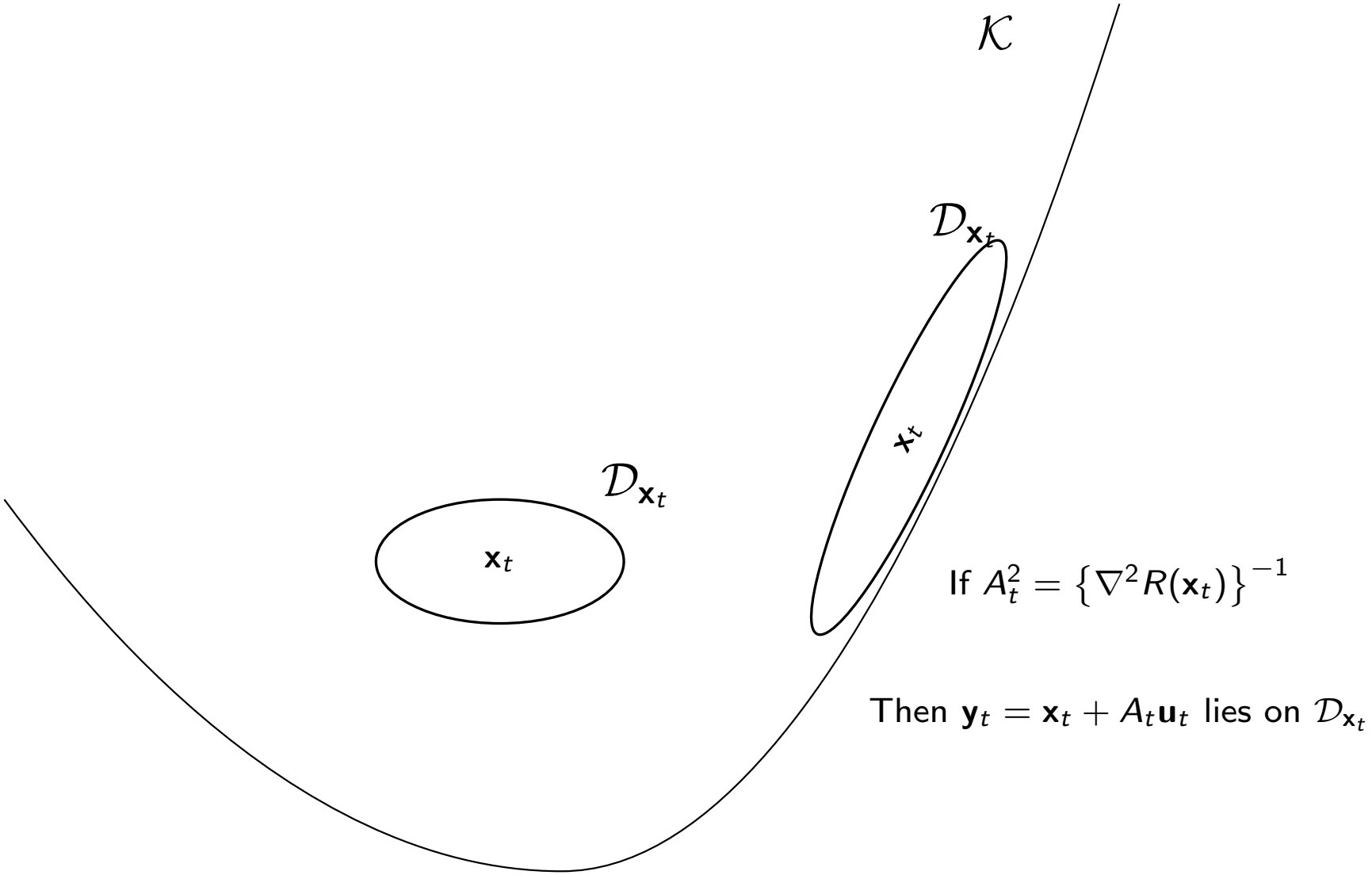

$\mathcal{K}$



- Dikin ellipsoid always lies inside $\mathcal{K}$.



- Dikin ellipsoid always lies inside $\mathcal{K}$.
- R needs to curve strongly near the boundary.



- Dikin ellipsoid always lies inside $\mathcal{K}$.
- R needs to curve strongly near the boundary.

Combining Ideas

- Combine single point gradient estimate with the idea of self concordant barriers.

Combining Ideas

- Combine single point gradient estimate with the idea of self concordant barriers.
- Removes the need for projections.

Combining Ideas

- Combine single point gradient estimate with the idea of self concordant barriers.
- Removes the need for projections.
- Generate gradient estimate $\mathbf{g}_t$ at step t.
- Feed it to the full information algorithm blackbox from [AHR08] to obtain

$$\mathbf{x}_{t+1} = \operatorname*{argmin}_{\mathbf{x} \in \mathcal{K}} \sum_{i=1}^{t} \eta \langle \mathbf{g}_i, \mathbf{x} \rangle + R(\mathbf{x})$$

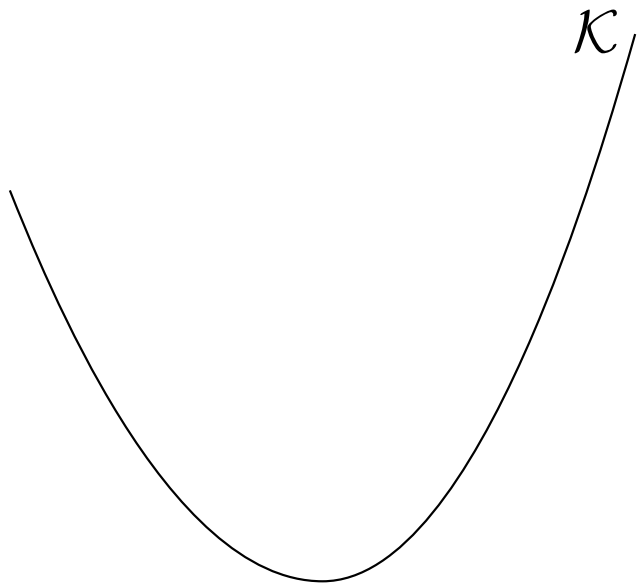
Algorithm

- Given R: a self concordant barrier for $\mathcal{K}$.

Algorithm

- Given R: a self concordant barrier for $\mathcal{K}$.
- Pick $\mathbf{x}_1 \in \mathcal{K}$.

At every step $t = 1, 2, 3, \ldots,$

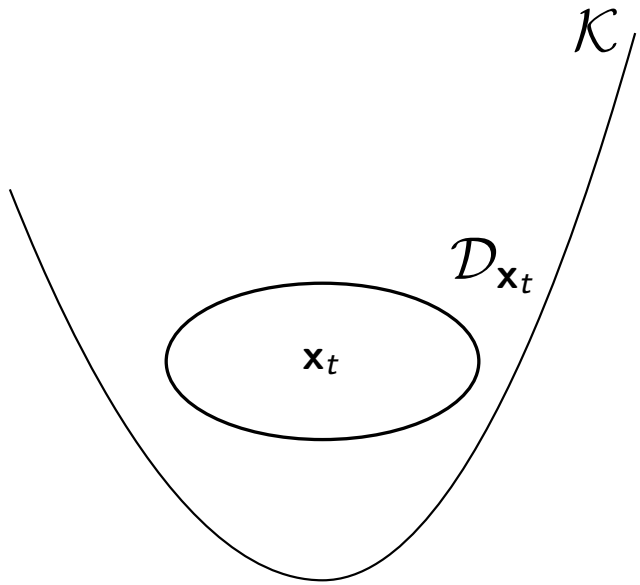


Algorithm

- Given R: a self concordant barrier for $\mathcal{K}$.
- Pick $\mathbf{x}_1 \in \mathcal{K}$.

At every step $t = 1, 2, 3, \ldots,$

- Evaluate $A_t = \left(\nabla^2 R(\mathbf{x}_t)\right)^{-1/2}$.

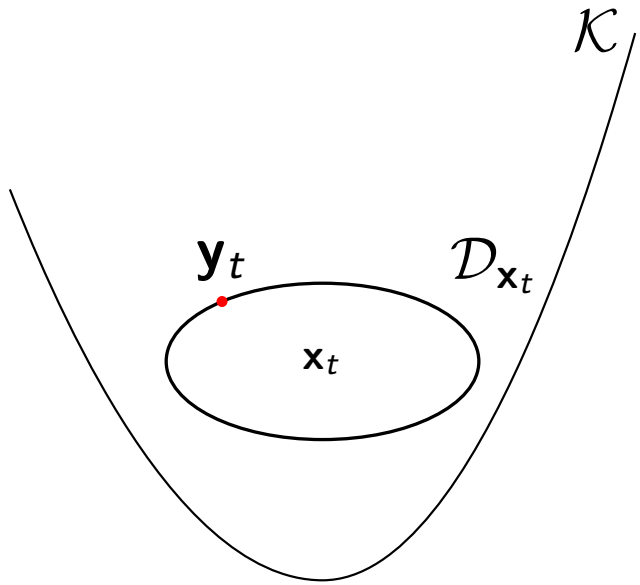


Algorithm

- Given R: a self concordant barrier for $\mathcal{K}$.
- Pick $\mathbf{x}_1 \in \mathcal{K}$.

At every step $t = 1, 2, 3, \ldots,$

- Evaluate $A_t = \left(\nabla^2 R(\mathbf{x}_t)\right)^{-1/2}$.
- Sample $\mathbf{u}_t \in \mathbb{S}^d$.
  $\mathbf{y}_t = \mathbf{x}_t + \delta A_t \mathbf{u}_t$.

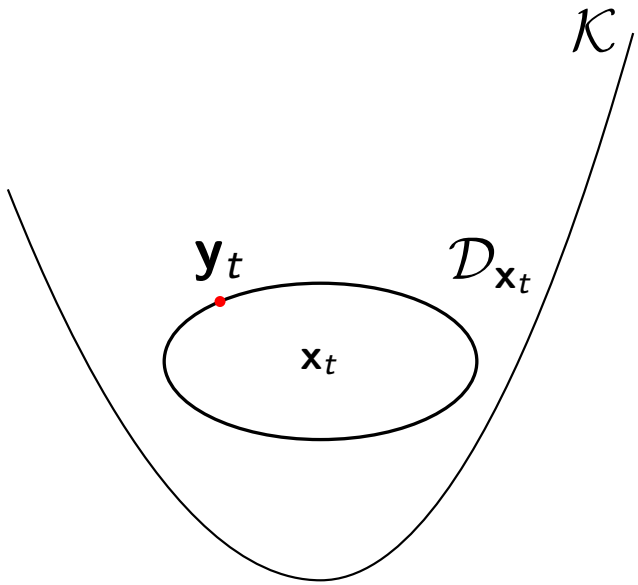


Algorithm

- Given R: a self concordant barrier for $\mathcal{K}$.
- Pick $\mathbf{x}_1 \in \mathcal{K}$.

At every step $t = 1, 2, 3, \ldots,$

- Evaluate $A_t = \left(\nabla^2 R(\mathbf{x}_t)\right)^{-1/2}$.
- Sample $\mathbf{u}_t \in \mathbb{S}^d$.
  $\mathbf{y}_t = \mathbf{x}_t + \delta A_t \mathbf{u}_t$.
- Player plays $\mathbf{y}_t$ and receives loss $f_t(\mathbf{y}_t)$.

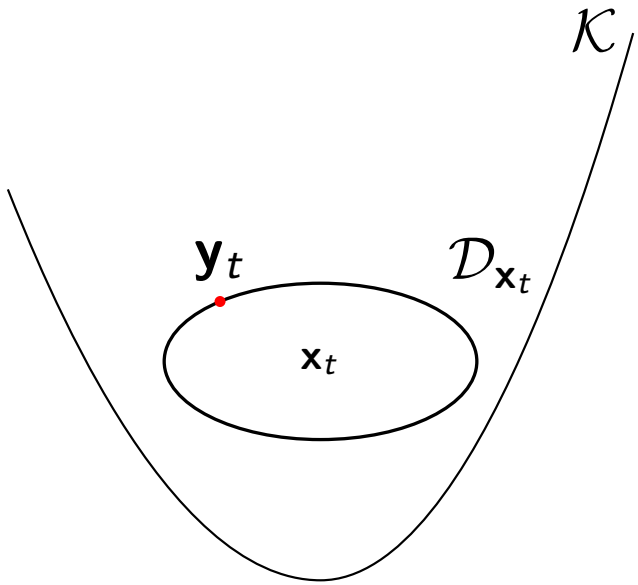


Algorithm

- Given R: a self concordant barrier for $\mathcal{K}$.
- Pick $\mathbf{x}_1 \in \mathcal{K}$.

At every step $t = 1, 2, 3, \ldots,$

- Evaluate $A_t = \left(\nabla^2 R(\mathbf{x}_t)\right)^{-1/2}$.
- Sample $\mathbf{u}_t \in \mathbb{S}^d$.
  $\mathbf{y}_t = \mathbf{x}_t + \delta A_t \mathbf{u}_t$.
- Player plays $\mathbf{y}_t$ and receives loss $f_t(\mathbf{y}_t)$.
- $\mathbf{g}_t = \frac{d}{\delta} f_t(\mathbf{y}_t) A_t^{-1} \mathbf{u}_t$.

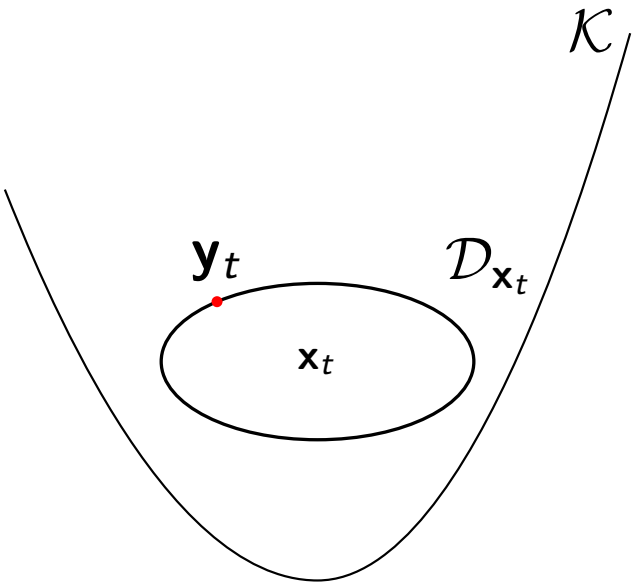


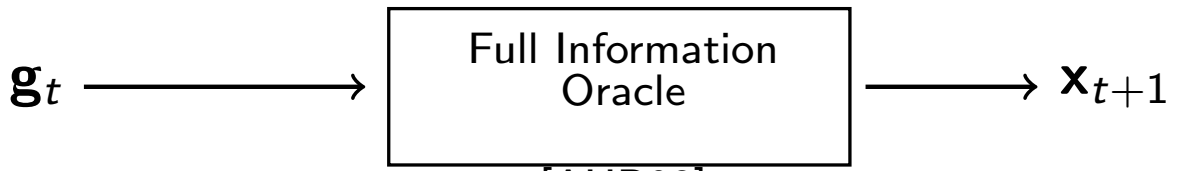
Algorithm

- Given R: a self concordant barrier for $\mathcal{K}$.
- Pick $\mathbf{x}_1 \in \mathcal{K}$.

At every step $t = 1, 2, 3, \ldots,$

- Evaluate $A_t = \left(\nabla^2 R(\mathbf{x}_t)\right)^{-1/2}$.
- Sample $\mathbf{u}_t \in \mathbb{S}^d$.
  $\mathbf{y}_t = \mathbf{x}_t + \delta A_t \mathbf{u}_t$.
- Player plays $\mathbf{y}_t$ and receives loss $f_t(\mathbf{y}_t)$.
- $\mathbf{g}_t = \frac{d}{\delta} f_t(\mathbf{y}_t) A_t^{-1} \mathbf{u}_t$.





[AHR08]

Proof Sketch

- Define the approximation

$$\hat{f}_t(\mathbf{x}) = \mathbb{E}_{\mathbf{v} \in \mathbb{B}^d} \left[f_t(\mathbf{x} + \delta A_t \mathbf{v}_t) \right]$$

- Regret written as a telescopic sum.
- Using approximation + Randomization incurs $O(T\delta^2)$ cost.
- [AHR08] blackbox incurs $O^*(\sqrt{T}/\delta)$.
- Trading off δ gives $O^*(T^{2/3})$ regret.

Conclusions and Future Work

- Improve bounds on regret for bandit OCO from $O(T^{3/4})$ to $O^*(T^{2/3})$, when f_t are smooth.

Conclusions and Future Work

- Improve bounds on regret for bandit OCO from $O(T^{3/4})$ to $O^*(T^{2/3})$, when f_t are smooth.
- Lower Bounds still correspond to full information setting.

Conclusions and Future Work

- Improve bounds on regret for bandit OCO from $O(T^{3/4})$ to $O^*(T^{2/3})$, when f_t are smooth.
- Lower Bounds still correspond to full information setting.
- How to bridge the gap?

Conclusions and Future Work

- Improve bounds on regret for bandit OCO from $O(T^{3/4})$ to $O^*(T^{2/3})$, when f_t are smooth.
- Lower Bounds still correspond to full information setting.
- How to bridge the gap?
- Better bounds on "price of bandit information" for more specific function classes?

Conclusions and Future Work

- Improve bounds on regret for bandit OCO from $O(T^{3/4})$ to $O^*(T^{2/3})$, when f_t are smooth.
- Lower Bounds still correspond to full information setting.
- How to bridge the gap?
- Better bounds on "price of bandit information" for more specific function classes?

Thank You!


Alekh Agarwal, Ofer Dekel, and Lin Xiao.
Optimal algorithms for online convex optimization with multi-point bandit feedback, 2010.
longer version available at
http://www.cs.berkeley.edu/~alekh/bandit-colt.pdf.


Jacob Abernethy, Elad Hazan, and Alexander Rakhlin.
Competing in the dark: An efficient algorithm for bandit linear optimization.
In *COLT*, pages 263–274, 2008.


Abraham Flaxman, Adam Tauman Kalai, and H. Brendan McMahan.
Online convex optimization in the bandit setting: gradient descent without a gradient.
In *SODA*, pages 385–394, 2005.


Elad Hazan, Amit Agarwal, and Satyen Kale.
Logarithmic regret algorithms for online convex optimization.
Machine Learning, 69(2-3):169–192, 2007.


M. Zinkevich.
Online convex programming and generalised infinitesimal gradient ascent.
In *ICML*, pages 928–936, 2003.