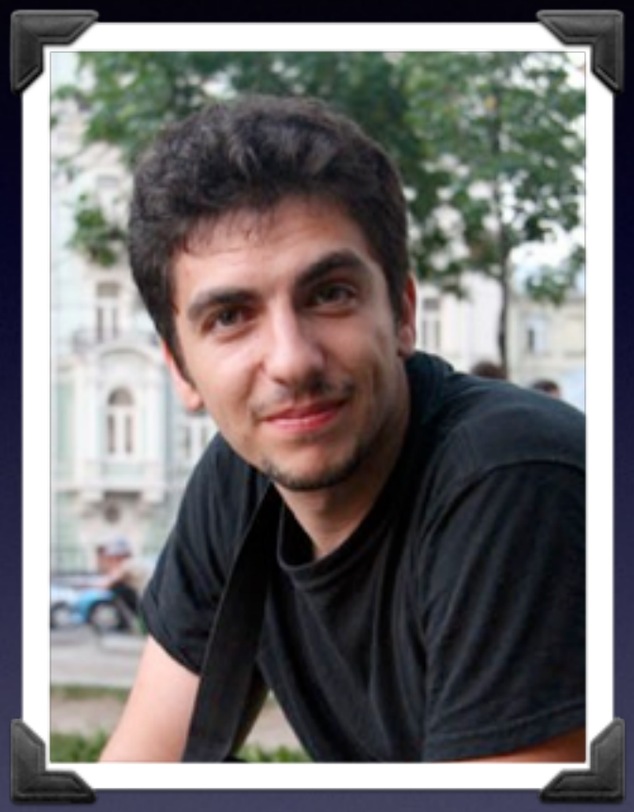
An Efficient Algorithm for Bandit Linear Optimization

Jacob Abernethy
UC Berkeley (PhD) =====> UPenn (Postdoc)
(in between jobs...)

Coauthors

Coauthors



Sasha Rakhlin
UPenn Statistics

Elad Hazan
Technion IEOR

Many thanks to my advisor Peter Bartlett

Act I: Online Linear Optimization

Online Linear Optimization (OLO)

Decision Space: $\mathcal{K} \overset{\text{cvx}}{\subset} \mathbb{R}^d$ compact, convex

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

- Player chooses $\mathbf{x}_t \in \mathcal{K}$
- Adversary chooses linear (convex) $\mathbf{f}_t(\cdot)$
- Player suffers $\mathbf{f}_t(\mathbf{x}_t)$ (or, $\mathbf{f}_t \cdot \mathbf{x}_t$)

What's The Objective?

$$\frac{1}{T}\sum_{t=1}^{T}\mathbf{f}_t(\mathbf{x}_t)$$

What's The Objective?

$$\frac{1}{T}\sum_{t=1}^{T}\mathbf{f}_t(\mathbf{x}_t) - \min_{\mathbf{x}^*\in\mathcal{K}} \mathop{\mathbb{E}}_{\mathbf{f}} \mathbf{f}(\mathbf{x}^*)$$

What's The Objective?

$$\underbrace{\sum_{t=1}^{T} \mathbf{f}_t(\mathbf{x}_t) - \min_{\mathbf{x}^* \in \mathcal{K}} \sum_{t=1}^{T} \mathbf{f}_t(\mathbf{x}^*)}$$

Regret

Why Minimize Regret? And Applications…

- **Modeling**: very few assumptions
- **Competitive Framework**: provides guarantees *relative* to a benchmark
- **Finance**: "Universal" portfolios [Cover '91]
- **Probability-One**: gives deterministic guarantees to the decision-maker, like options contracts [DeMarzo et al. '06]
- **Analytical Tool**: is used to study sequential optimization, and e.g. convergence of game play to equilibrium

Warmup:
Two Sequential Decision Problems

Example 1: The "Expert Setting"

- There are a set of N experts (or "actions")
- For all t:
  - Player chooses distribution $\mathbf{w}_t$ over experts
  - The experts' costs $\mathbf{c}_t$ are revealed
  - Player suffers expected cost $\mathbf{w}_t \cdot \mathbf{c}_t$
- Want to minimize $\sum_t \mathbf{w}_t \cdot \mathbf{c}_t - \min_i \sum_t \mathbf{c}_t(i)$
- Typical Alg: $\mathbf{w}_{t+1}(i) \propto \mathbf{w}_t(i) \exp(\mathbf{c}_t(i))$

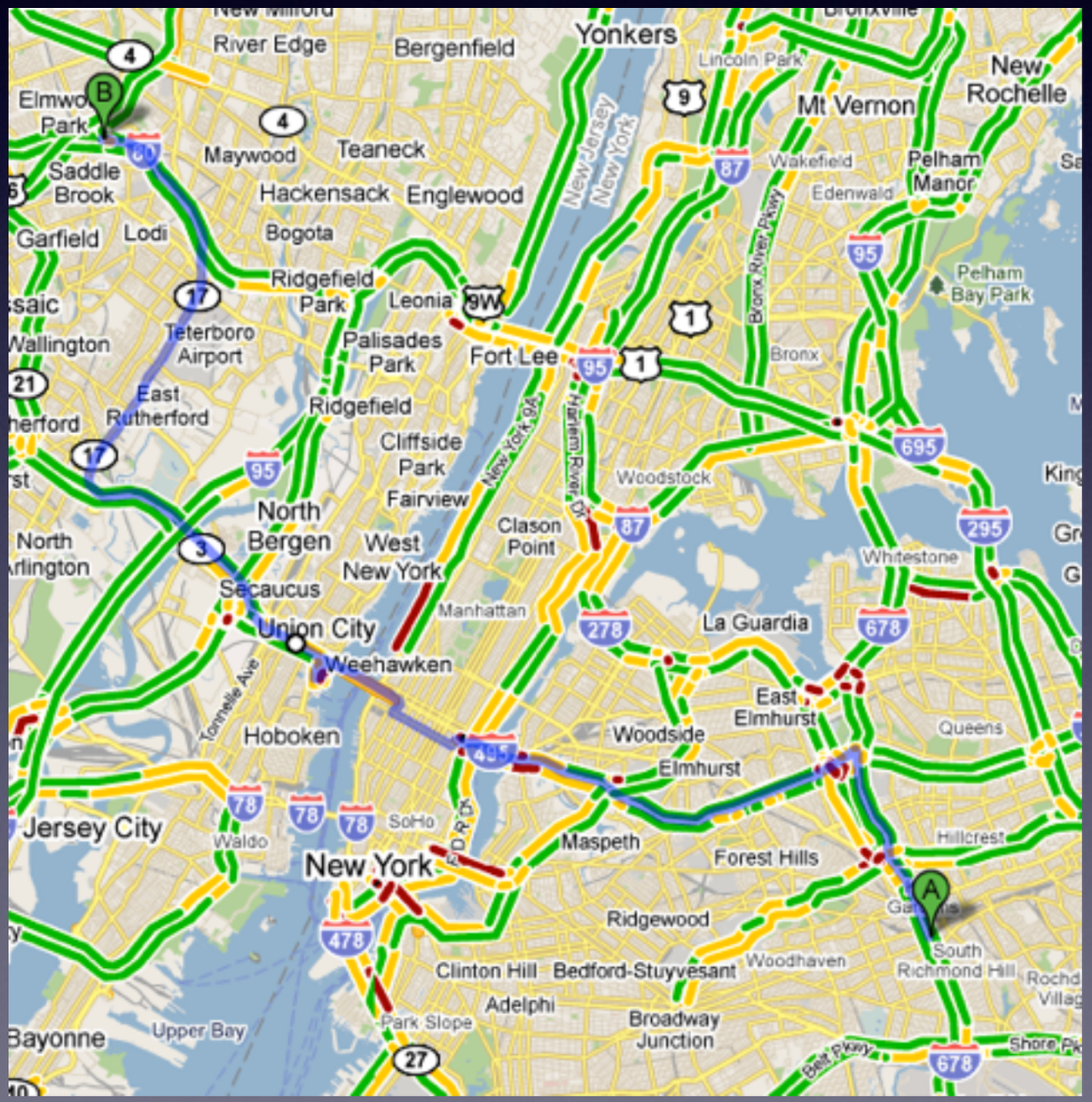
Example 2:
Driving to Work

Example 2: Driving to Work

Example 2: Driving to Work

Example 2: Driving to Work

Example 2: Driving to Work



Efficient Approach: Flow Polytope

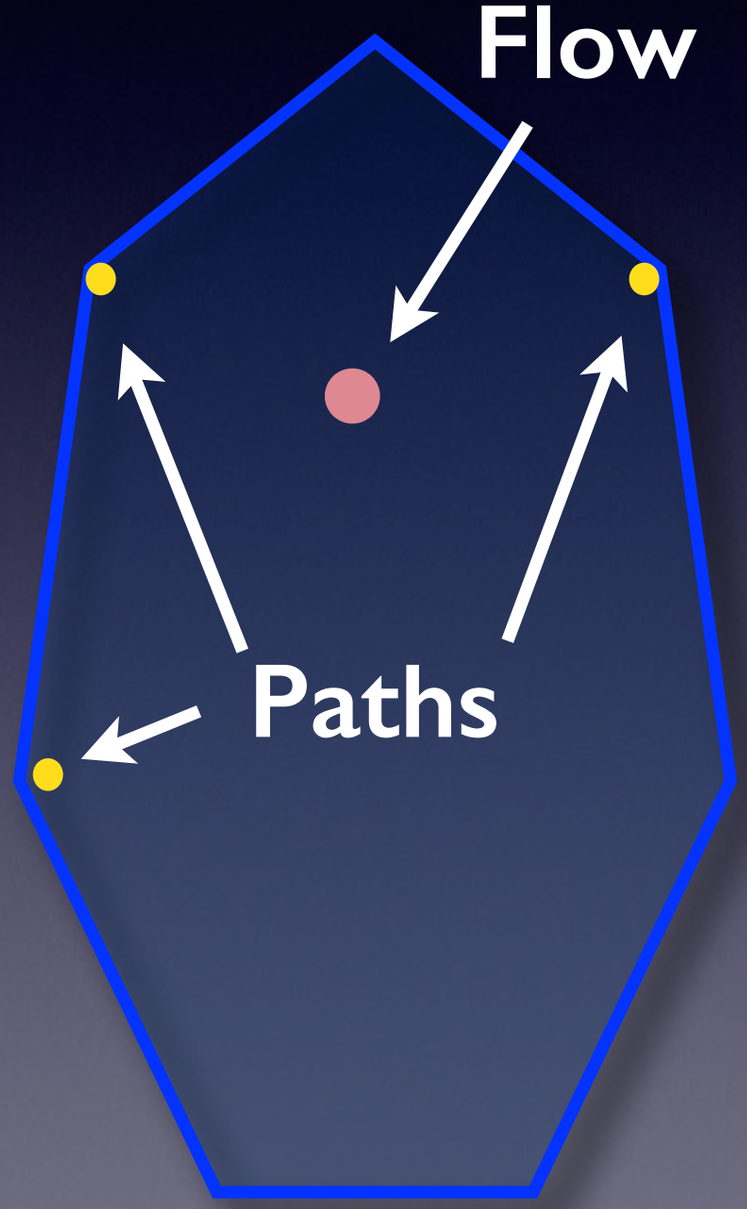
- **Flow Polytope:** convex hull of all paths from Source to Sink
- Problem dim = # edges
- Efficient decomposition from flows to randomized paths

Efficient Approach: Flow Polytope

- **Flow Polytope:** convex hull of all paths from Source to Sink
- Problem dim = # edges
- Efficient decomposition from flows to randomized paths

Efficient Approach: Flow Polytope

- **Flow Polytope:** convex hull of all paths from Source to Sink
- Problem dim = # edges
- Efficient decomposition from flows to randomized paths



Online Linear Optimization

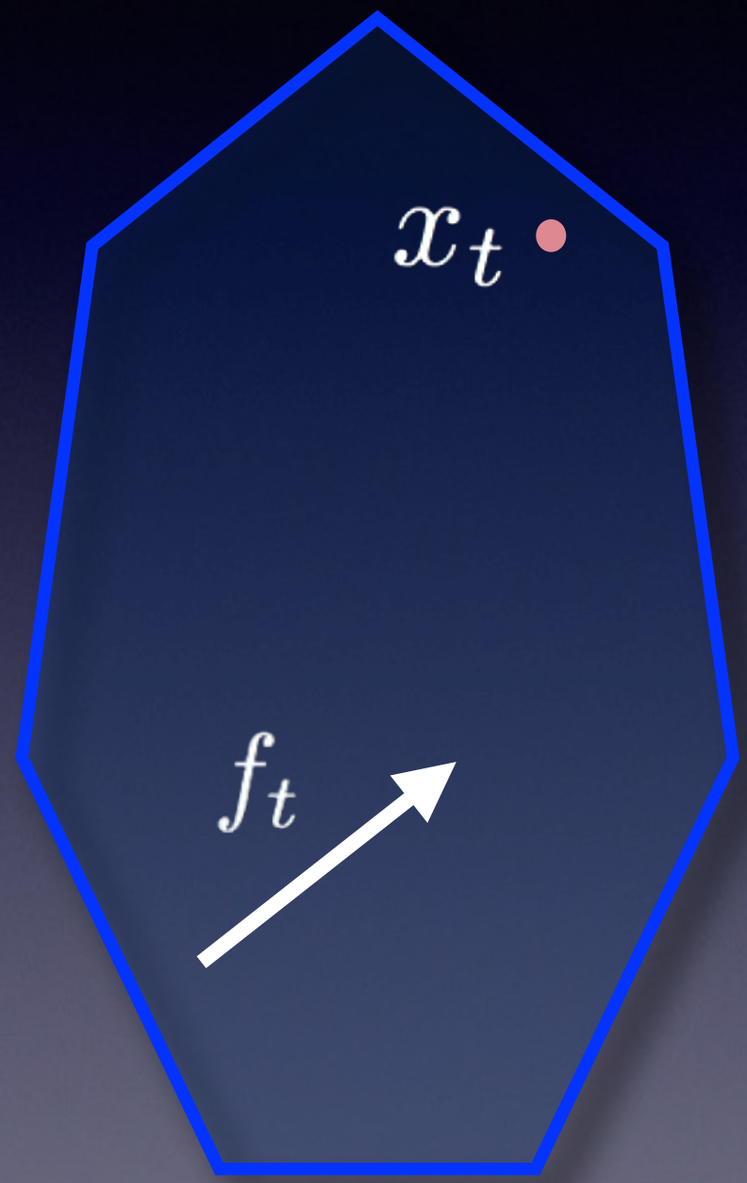
- On each round t:

Online Linear Optimization

- On each round t:
  - Player chooses x_t in D, a convex polytope (e.g. the "flow space" of the network)

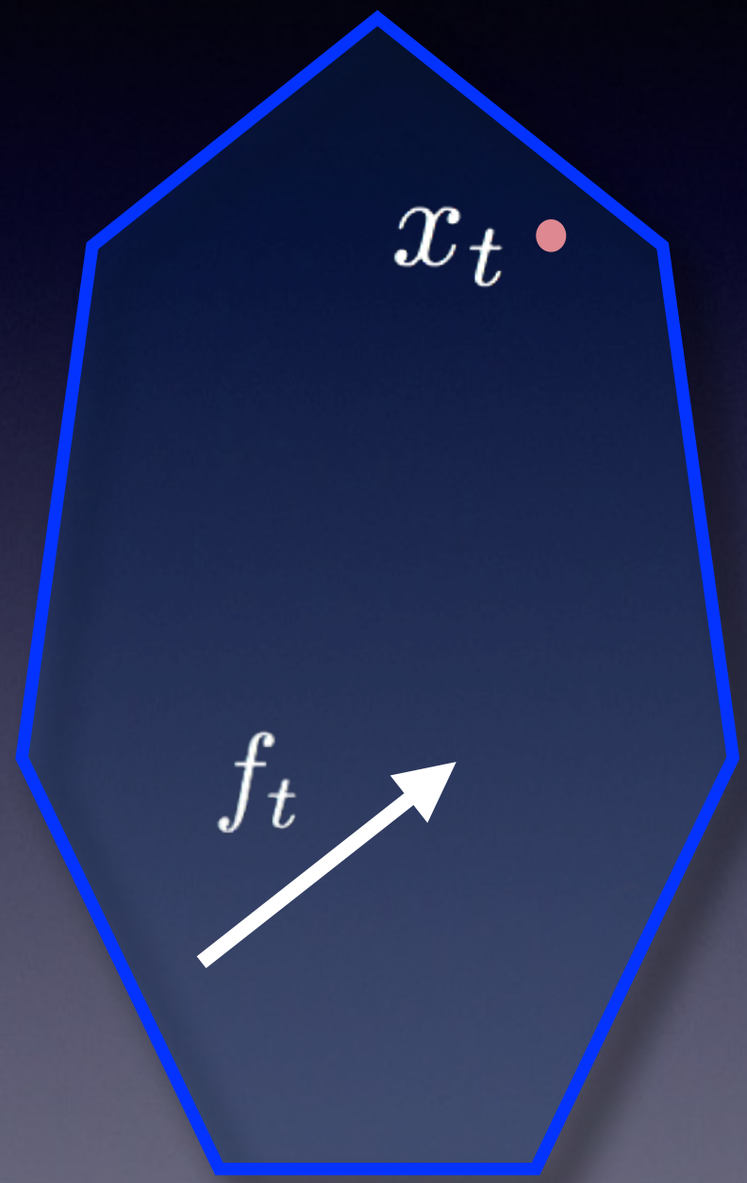
Online Linear Optimization

- On each round t:
  - Player chooses x_t in D, a convex polytope (e.g. the "flow space" of the network)
  - Adversary chooses f_t, a linear function (e.g. the vector of traffic times)



Online Linear Optimization

- On each round t:
  - Player chooses x_t in D, a convex polytope (e.g. the "flow space" of the network)
  - Adversary chooses f_t, a linear function (e.g. the vector of traffic times)
  - Player pays $f_t \cdot x_t$, *the "expected" cost*



Tight Regret Bounds for "Full Info" setting

Regret: $$\sum_{t=1}^{T} \mathbf{f}_t(\mathbf{x}_t) - \min_{\mathbf{x}^* \in \mathcal{K}} \sum_{t=1}^{T} \mathbf{f}_t(\mathbf{x}^*)$$

Tight Regret Bounds for "Full Info" setting

Setting	Regret rate
$\mathcal{K} := \Delta_d, \mathbf{f}_t \in [0,1]^d$	$\sqrt{T \log d}$
$\mathcal{K} := B_2, \mathbf{f}_t \in B_2$	$\sqrt{T}$
$\mathcal{K} := [0,1]^d, \mathbf{f}_t \in [0,1]^d$	$d\sqrt{T}$
$\mathcal{K} := B_2, \mathbf{f}_t \in [0,1]^d$	$\sqrt{dT}$
$\nabla \mathbf{f}_t \in B_2, \nabla^2 \mathbf{f}_t \succeq \alpha I$	$\frac{\log T}{\alpha}$

Regret: $$\sum_{t=1}^{T} \mathbf{f}_t(\mathbf{x}_t) - \min_{\mathbf{x}^* \in \mathcal{K}} \sum_{t=1}^{T} \mathbf{f}_t(\mathbf{x}^*)$$

The $T^{1/2}$ Regret Barrier

The $T^{1/2}$ Regret Barrier

- Decision set is $[-1, 1]$
- Adversary randomly samples $f_t \in \{-1, 1\}$
- For any player strategy:

$$\begin{aligned} \text{Worst case regret} &\geq \mathbb{E}\left[\sum f_t \cdot x_t - \min_{x \in [-1,1]} \sum f_t \cdot x\right] \\ &= 0 + \mathbb{E}\left|\sum f_t\right| \\ &= \Theta(\sqrt{T}) \end{aligned}$$

The $T^{1/2}$ Regret Barrier

- Decision set is $[-1, 1]$
- Adversary randomly samples $f_t \in \{-1, 1\}$
- For any player strategy:

$$\begin{aligned} \text{Worst case regret} &\geq \mathbb{E}\left[\sum f_t \cdot x_t - \min_{x \in [-1,1]} \sum f_t \cdot x\right] \\ &= 0 + \mathbb{E}\left|\sum f_t\right| \\ &= \Theta(\sqrt{T}) \end{aligned}$$

Conclusion: at least $T^{1/2}$ regret is unavoidable

Algorithm 1: Follow the Leader

$$\mathbf{x}_t := \arg\min_{\mathbf{x}\in\mathcal{K}} \sum_{s=1}^{t-1} \mathbf{f}_s \cdot \mathbf{x}$$

Algorithm 2: Follow the *Regularized* Leader (FTRL)

$$\mathbf{x}_t := \arg\min_{\mathbf{x}\in\mathcal{K}} \sum_{s=1}^{t-1} \mathbf{f}_s \cdot \mathbf{x} + \lambda R(\mathbf{x})$$

Algorithm 2: Follow the *Regularized* Leader (FTRL)

"Regularize" with a "curved" convex function

$$\mathbf{x}_t := \arg\min_{\mathbf{x}\in\mathcal{K}} \sum_{s=1}^{t-1} \mathbf{f}_s \cdot \mathbf{x} + \lambda R(\mathbf{x})$$

Algorithm 2: Follow the *Regularized* Leader (FTRL)

"Regularize" with a "curved" convex function

$$\mathbf{x}_t := \arg\min_{\mathbf{x}\in\mathcal{K}} \sum_{s=1}^{t-1} \mathbf{f}_s \cdot \mathbf{x} + \lambda R(\mathbf{x})$$

Choices of R leads to different algs:

$R(\mathbf{x}) = \sum x_i \log x_i \implies$ "Exponentiated Gradient" alg

$R(\mathbf{x}) = \|\mathbf{x}\|_2^2 \implies$ "Online Gradient Descent" alg

Useful Property: Strong Convexity

$$\underbrace{R(x) - R(y) - \nabla R(y) \cdot (x - y)}_{D_R(x,y)} \geq \frac{\|x - y\|^2}{2}$$

$D_R(x, y)$ ("Bregman" divergence of R)

Useful Property: Strong Convexity

$$\underbrace{R(x) - R(y) - \nabla R(y) \cdot (x - y)}_{D_R(x,y)} \geq \frac{\|x - y\|^2}{2}$$

(“Bregman” divergence of R)

$$R(x) = \|x\|_2^2 \implies D_R(x,y) = \|x - y\|_2^2$$

$$R(x) = \sum_i x_i \log x_i \implies D_R(x,y) = \sum x_i \log \frac{x_i}{y_i}$$

FTRL Regret Bound

$$\text{Regret}_T \leq \sum_{t=1}^{T} \lambda D_R(\mathbf{x}_t, \mathbf{x}_{t+1}) + \lambda R(\mathbf{x}^*)$$

FTRL Regret Bound

$$\text{Regret}_T \leq \sum_{t=1}^{T} \lambda D_R(\mathbf{x}_t, \mathbf{x}_{t+1}) + \lambda R(\mathbf{x}^*)$$

Strong Convexity! →

$$\leq \sum_{t=1}^{T} \frac{\|\mathbf{f}_t\|_*^2}{\lambda} + \lambda R(\mathbf{x}^*)$$

FTRL Regret Bound

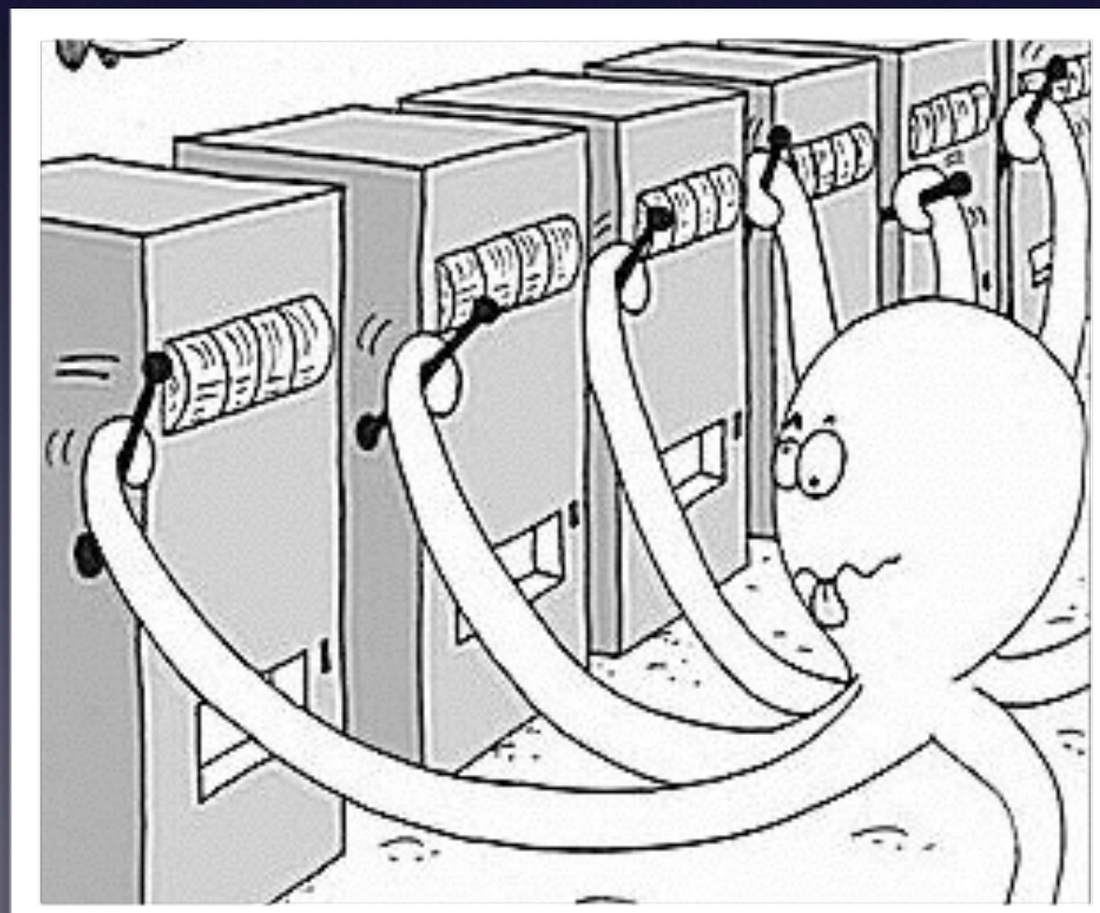
$$\text{Regret}_T \leq \sum_{t=1}^{T} \lambda D_R(\mathbf{x}_t, \mathbf{x}_{t+1}) + \lambda R(\mathbf{x}^*)$$

Strong Convexity! →

$$\leq \sum_{t=1}^{T} \frac{\|\mathbf{f}_t\|_*^2}{\lambda} + \lambda R(\mathbf{x}^*)$$

$$\leq \frac{T \cdot G}{\lambda} + \lambda D \leq 2\sqrt{T \cdot G \cdot D}$$

Online Linear Opt. in the "Bandit" Setting



OLO for *Full Info* and *Bandit* Feedback

- Two feedback settings:

- **Full-Information**:
  Player sees the full loss function f_t

- **Bandit**:
  Player sees cost of chosen action $f_t \cdot x_t$

FTRL in Bandit Setting?

$$\mathbf{x}_t := \arg\min_{\mathbf{x}\in\mathcal{K}} \sum_{s=1}^{t-1} \mathbf{f}_s \cdot \mathbf{x} + \lambda R(\mathbf{x})$$

FTRL in Bandit Setting?

$$\mathbf{x}_t := \arg\min_{\mathbf{x}\in\mathcal{K}} \sum_{s=1}^{t-1} \mathbf{f}_s \cdot \mathbf{x} + \lambda R(\mathbf{x})$$

Major Bug:
we never observe this vector

Solution: One-sample Estimates

- Can we really get an unbiased estimate from one sample? Yes!

Solution: One-sample Estimates

- Can we really get an unbiased estimate from one sample? Yes!

- Input: hidden vector $\ell := \langle \ell_1, \ldots, \ell_n \rangle$
- Sample $I \in \{1, \ldots, n\}$ uniformly at random
- Witness random coordinate ℓ_I
- Output: estimate $\hat{\ell} := \langle 0, \ldots, 0, n\ell_I, 0, \ldots, 0 \rangle$

One-Sample Estimates Part II

- In previous example, I had to sample uniformly at random from coordinates
- BUT I want low regret, and ALG tells me to sample according to **p**. OK, modification:

One-Sample Estimates Part II

- In previous example, I had to sample uniformly at random from coordinates
- BUT I want low regret, and ALG tells me to sample according to **p**. OK, modification:
  - Sample $I \in \{1, \ldots, n\}$ according to $\mathbf{p}$
  - Witness random coordinate ℓ_I
  - Output: estimate $\hat{\ell} := \langle 0, \ldots, 0, \frac{\ell_I}{\mathbf{p}(I)}, 0, \ldots, 0 \rangle$

One-Sample Estimates Part II

- In previous example, I had to sample uniformly at random from coordinates

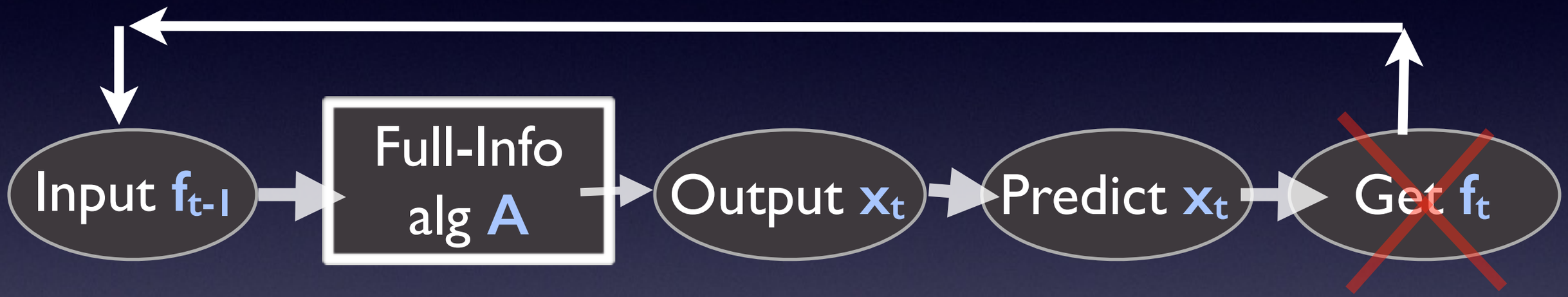
- BUT I want low regret, and ALG tells me to sample according to **p**. OK, modification:

  - Sample $I \in \{1, \ldots, n\}$ according to $\mathbf{p}$
  - Witness random coordinate ℓ_I
  - Output: estimate $\hat{\ell} := \langle 0, \ldots, 0, \frac{\ell_I}{\mathbf{p}(I)}, 0, \ldots, 0 \rangle$

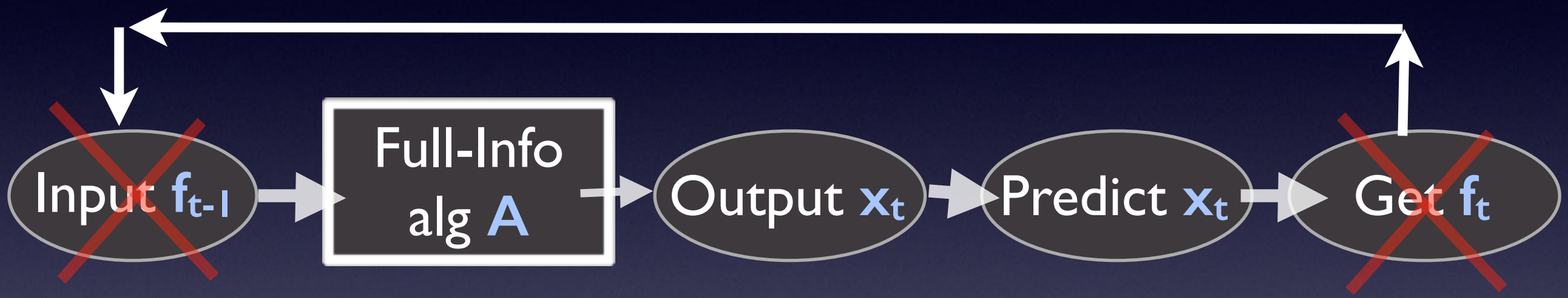
This Gets Big!!

Full Information to Bandit Feedback

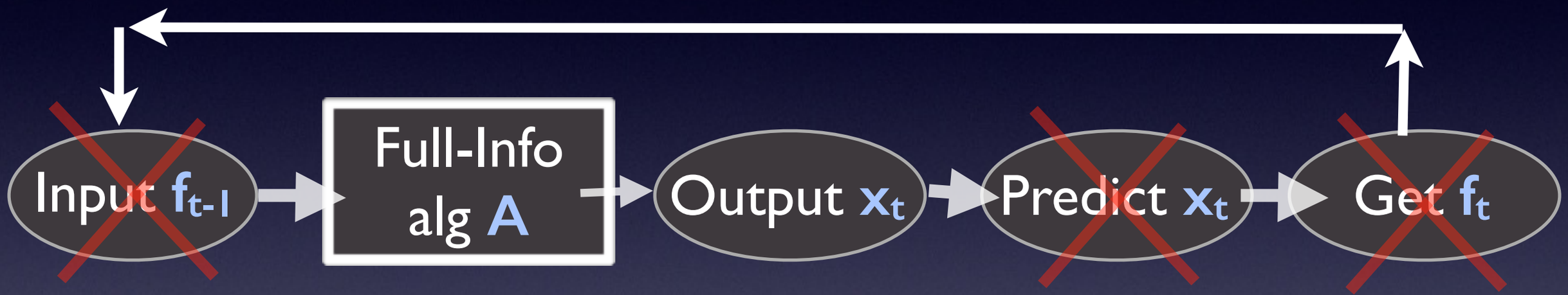
Full Information to Bandit Feedback



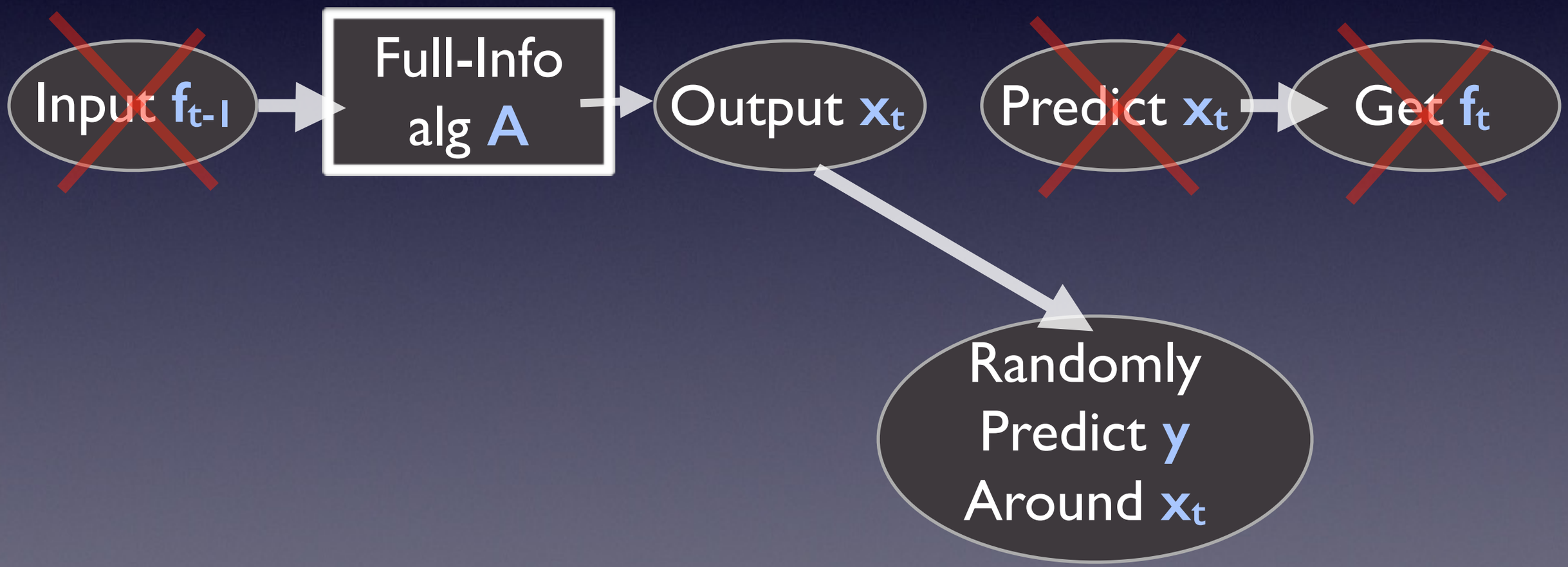
Full Information to Bandit Feedback



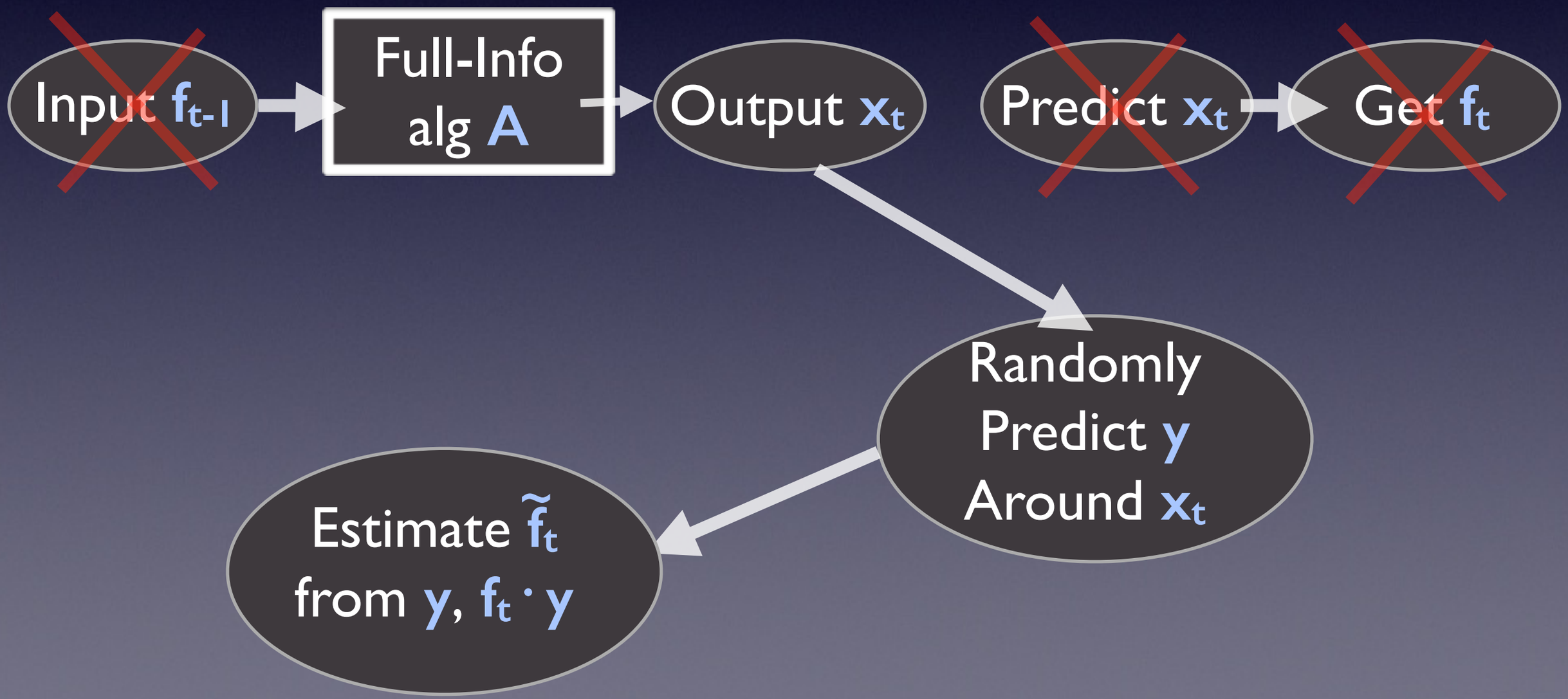
Full Information to Bandit Feedback



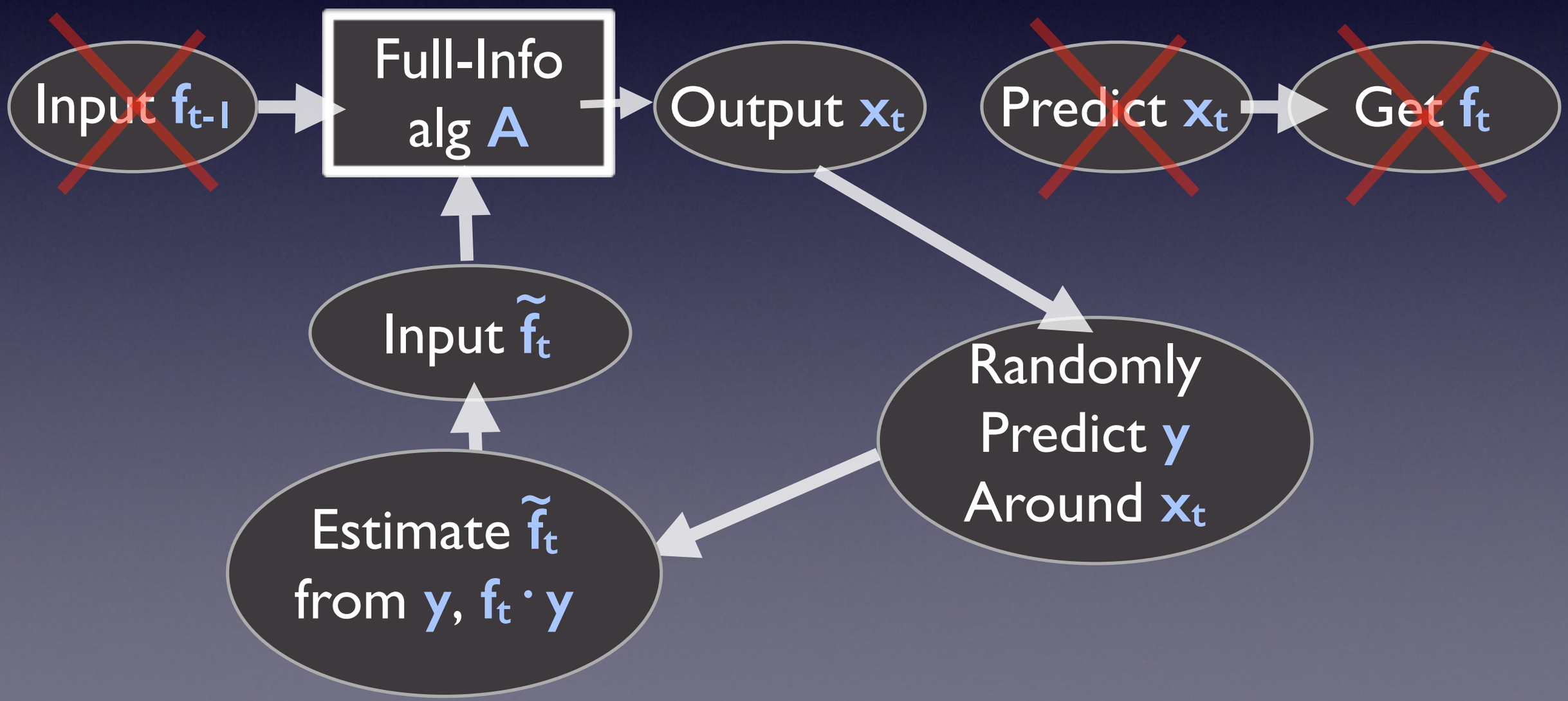
Full Information to Bandit Feedback



Full Information to Bandit Feedback



Full Information to Bandit Feedback



FTRL in Bandit Setting?

$$\mathbf{x}_t := \arg\min_{\mathbf{x}\in\mathcal{K}} \sum_{s=1}^{t-1} \mathbf{f}_s \cdot \mathbf{x} + \lambda R(\mathbf{x})$$

FTRL in Bandit Setting?

$$\mathbf{x}_t := \arg\min_{\mathbf{x}\in\mathcal{K}} \sum_{s=1}^{t-1} \tilde{\mathbf{f}}_s \cdot \mathbf{x} + \lambda R(\mathbf{x})$$

Solution:

Use the estimated functions!

Estimates are Enough?

Fortunately, it is sufficient to compete with an unbiased estimate of the loss functions

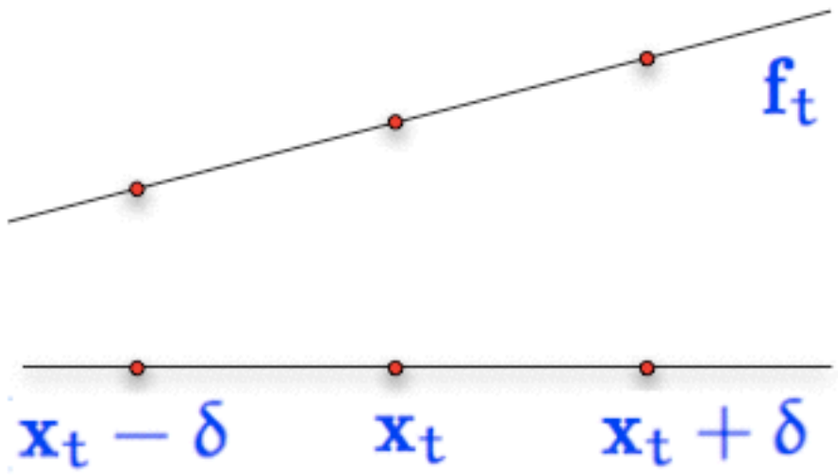
Estimates are Enough?

Fortunately, it is sufficient to compete with an unbiased estimate of the loss functions

$$\begin{aligned} \mathbb{E}\,\mathrm{Regret}(\tilde{f}_1, \ldots, \tilde{f}_T) &\geq \mathrm{Regret}(\mathbb{E}\tilde{f}_1, \ldots, \mathbb{E}\tilde{f}_T) \\ &= \mathrm{Regret}(f_1, \ldots, f_T) \end{aligned}$$

Easy to check: Regret is convex in f_t's

The Problem with One-Sample Estimates



Predict randomly $\mathbf{y}_t = \mathbf{x}_t \pm \delta$.

Set $\tilde{\mathbf{f}}_t = \pm(\mathbf{f}_t \cdot \mathbf{y}_t)/\delta$.

Unbiased:

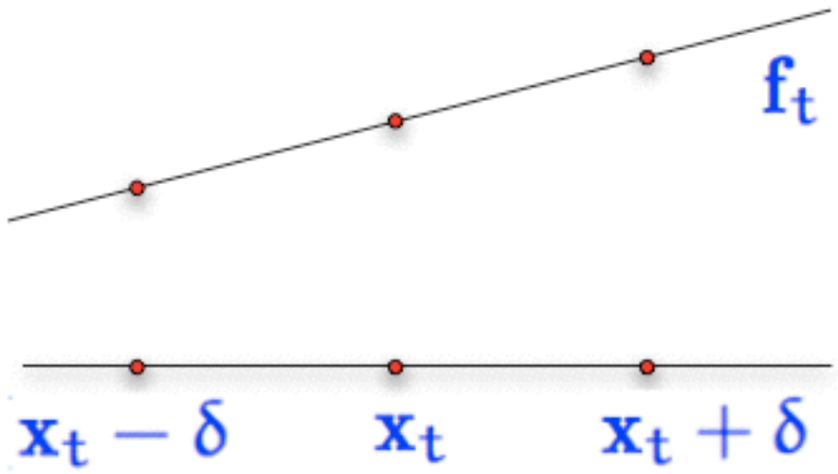
$$\mathbb{E}\tilde{\mathbf{f}}_t = \frac{1}{2}\frac{\mathbf{f}_t \cdot (\mathbf{x}_t + \delta)}{\delta} - \frac{1}{2}\frac{\mathbf{f}_t \cdot (\mathbf{x}_t - \delta)}{\delta} = \mathbf{f}_t$$

and

$$\mathbb{E}\mathbf{y}_t = \mathbf{x}_t$$

The Problem with One-Sample Estimates

Size of Estimate
≈
Distance to Boundary



Set $\tilde{\mathbf{f}}_t = \pm(\mathbf{f}_t \cdot \mathbf{y}_t)/\delta$.

Unbiased:

$$\mathbb{E}\tilde{\mathbf{f}}_t = \frac{1}{2}\frac{\mathbf{f}_t \cdot (\mathbf{x}_t + \delta)}{\delta} - \frac{1}{2}\frac{\mathbf{f}_t \cdot (\mathbf{x}_t - \delta)}{\delta} = \mathbf{f}_t$$

and

$$\mathbb{E}\mathbf{y}_t = \mathbf{x}_t$$

Predict randomly $\mathbf{y}_t = \mathbf{x}_t \pm \delta$.

FTRL Regret Bound

$$\text{Regret}_T \leq \sum_{t=1}^{T} \lambda D_R(\mathbf{x}_t, \mathbf{x}_{t+1}) + \lambda R(\mathbf{x}^*)$$

$$\leq \sum_{t=1}^{T} \frac{\|\tilde{\mathbf{f}}_t\|_*^2}{\lambda} + \lambda R(\mathbf{x}^*)$$

$$\leq \frac{T \cdot G}{\lambda} + \lambda D \leq 2\sqrt{T \cdot G \cdot D}$$

FTRL Regret Bound

$$\text{Regret}_T \leq \sum_{t=1}^{T} \lambda D_R(\mathbf{x}_t, \mathbf{x}_{t+1}) + \lambda R(\mathbf{x}^*)$$

Grows large!

$$\leq \sum_{t=1}^{T} \frac{\|\tilde{\mathbf{f}}_t\|_*^2}{\lambda} + \lambda R(\mathbf{x}^*)$$

$$\leq \frac{T \cdot G}{\lambda} + \lambda D \leq 2\sqrt{T \cdot G \cdot D}$$

Timeline of Results on Adversarial Bandit

	Regret	Efficient?	Convex Sets?	W.H.P?
Auer, Cesa-Bianchi, Freund, Schapire 02	$\sqrt{T}$	✓	Simplex	✓

Timeline of Results on Adversarial Bandit

	Regret	Efficient?	Convex Sets?	W.H.P?
Auer, Cesa-Bianchi, Freund, Schapire 02	$\sqrt{T}$	✓	Simplex	✓
Awerbuch and Kleinberg 04	$T^{2/3}$	✓	Flows	✗

Timeline of Results on Adversarial Bandit

	Regret	Efficient?	Convex Sets?	W.H.P?
Auer, Cesa-Bianchi, Freund, Schapire 02	$\sqrt{T}$	✓	Simplex	✓
Awerbuch and Kleinberg 04	$T^{2/3}$	✓	Flows	✗
McMahan and Blum 04	$T^{3/4}$	✓	All	✗
Flaxman, Kalai, McMahon 05	$T^{3/4}$	✓	All	✗
Gyorgy et al 07	$T^{2/3}$	✓	Flows	✓

Timeline of Results on Adversarial Bandit

	Regret	Efficient?	Convex Sets?	W.H.P?
Auer, Cesa-Bianchi, Freund, Schapire 02	$\sqrt{T}$	✓	Simplex	✓
Awerbuch and Kleinberg 04	$T^{2/3}$	✓	Flows	✗
McMahan and Blum 04	$T^{3/4}$	✓	All	✗
Flaxman, Kalai, McMahon 05	$T^{3/4}$	✓	All	✗
Gyorgy et al 07	$T^{2/3}$	✓	Flows	✓
Dani, Hayes, Kakade 07	$\sqrt{T}$	✗	All	✗

Timeline of Results on Adversarial Bandit

	Regret	Efficient?	Convex Sets?	W.H.P?
Auer, Cesa-Bianchi, Freund, Schapire 02	$\sqrt{T}$	✓	Simplex	✓
Awerbuch and Kleinberg 04	$T^{2/3}$	✓	Flows	✗
McMahan and Blum 04	$T^{3/4}$	✓	All	✗
Flaxman, Kalai, McMahon 05	$T^{3/4}$	✓	All	✗
Gyorgy et al 07	$T^{2/3}$	✓	Flows	✓
Dani, Hayes, Kakade 07	$\sqrt{T}$	✗	All	✗
Abernethy, Hazan, Rakhlin 08	$\sqrt{T}$	✓	All	✗

Timeline of Results on Adversarial Bandit

	Regret	Efficient?	Convex Sets?	W.H.P?
Auer, Cesa-Bianchi, Freund, Schapire 02	$\sqrt{T}$	✓	Simplex	✓
Awerbuch and Kleinberg 04	$T^{2/3}$	✓	Flows	✗
McMahan and Blum 04	$T^{3/4}$	✓	All	✗
Flaxman, Kalai, McMahon 05	$T^{3/4}$	✓	All	✗
Gyorgy et al 07	$T^{2/3}$	✓	Flows	✓
Dani, Hayes, Kakade 07	$\sqrt{T}$	✗	All	✗
Abernethy, Hazan, Rakhlin 08	$\sqrt{T}$	✓	All	✗
Abernethy and Rakhlin 09	$\sqrt{T}$	✓	All	✓

Timeline of Results on Adversarial Bandit

	Regret	Efficient?	Convex Sets?	W.H.P?
Auer, Cesa-Bianchi, Freund, Schapire 02	$\sqrt{T}$	✓	Simplex	✓
Awerbuch and Kleinberg 04	$T^{2/3}$	✓	Flows	✗
McMahan and Blum 04	$T^{3/4}$	✓	All	✗
Flaxman, Kalai, McMahon 05	$T^{3/4}$	✓	All	✗
Gyorgy et al 07	$T^{2/3}$	✓	Flows	✓
Dani, Hayes, Kakade 07	$\sqrt{T}$	✗	All	✗
Abernethy, Hazan, Rakhlin 08	$\sqrt{T}$	✓	All	✗
Abernethy and Rakhlin 09	$\sqrt{T}$	✓	All	✓

Two Crucial Dilemmas

Two Crucial Dilemmas

- How to sample around x_t and simultaneously construct unbiased estimates?

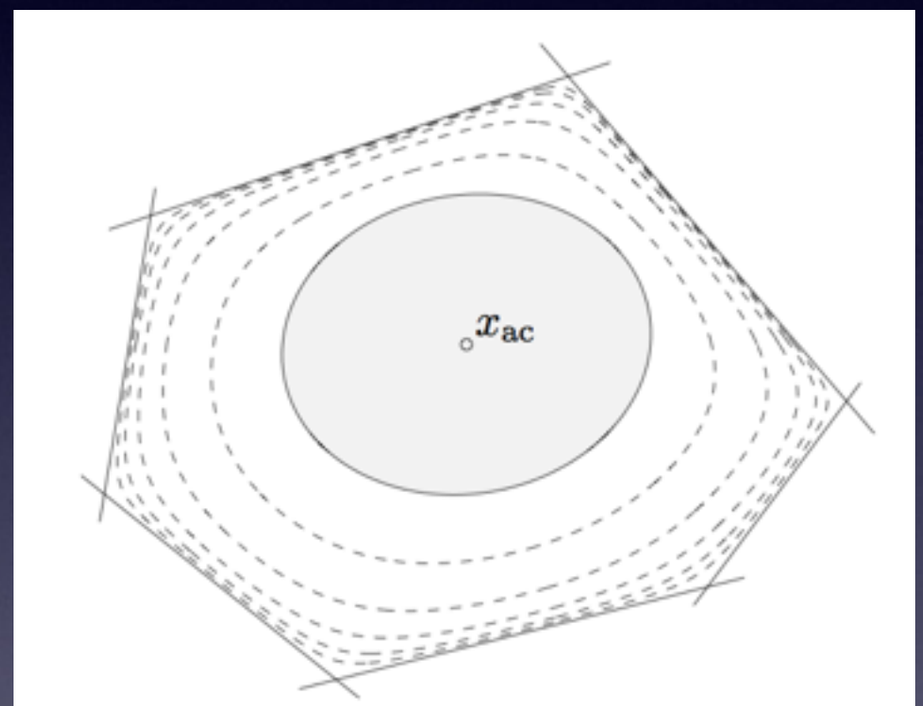
Two Crucial Dilemmas

- How to sample around x_t and simultaneously construct unbiased estimates?

- How to deal with estimates that get very large?

The Magic Trick: Self-Concordant Functions

- Long-standing open question: How to use **Newton's Method** for poly-time convex optimization?
- Nesterov and Nemirovsky: add "self-concordant barrier" to objective
- Restrictions on 2nd + 3rd derivatives
- Only efficient method for general convex optimization



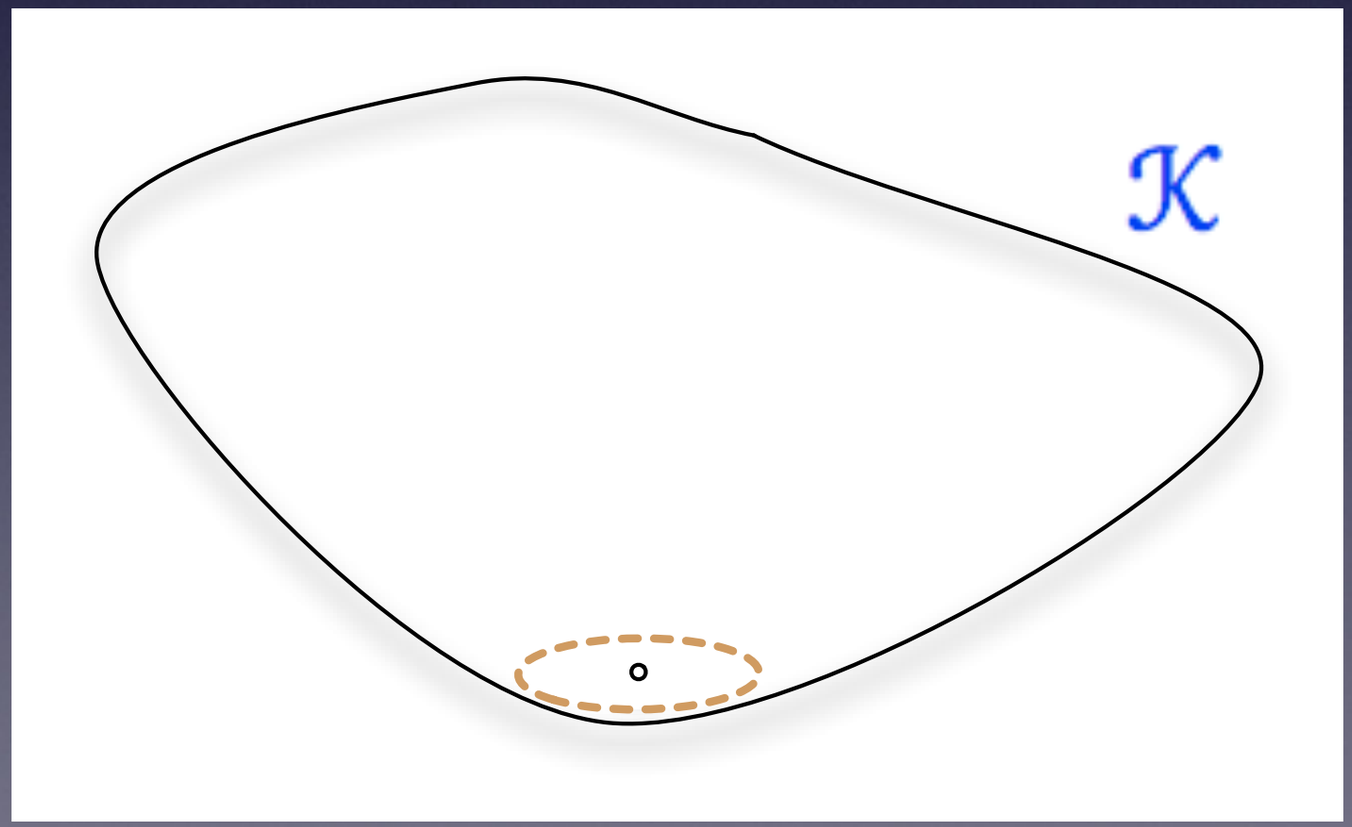
(Self-concordant is roughly "log-barrier")

A Bonus Prize: The Dikin Ellipsoid

$$\{\mathbf{z} \ : \ (\mathbf{z} - \mathbf{x}_t)\nabla^2\mathcal{R}(\mathbf{x}_t)(\mathbf{z} - \mathbf{x}_t) \leq 1\}$$

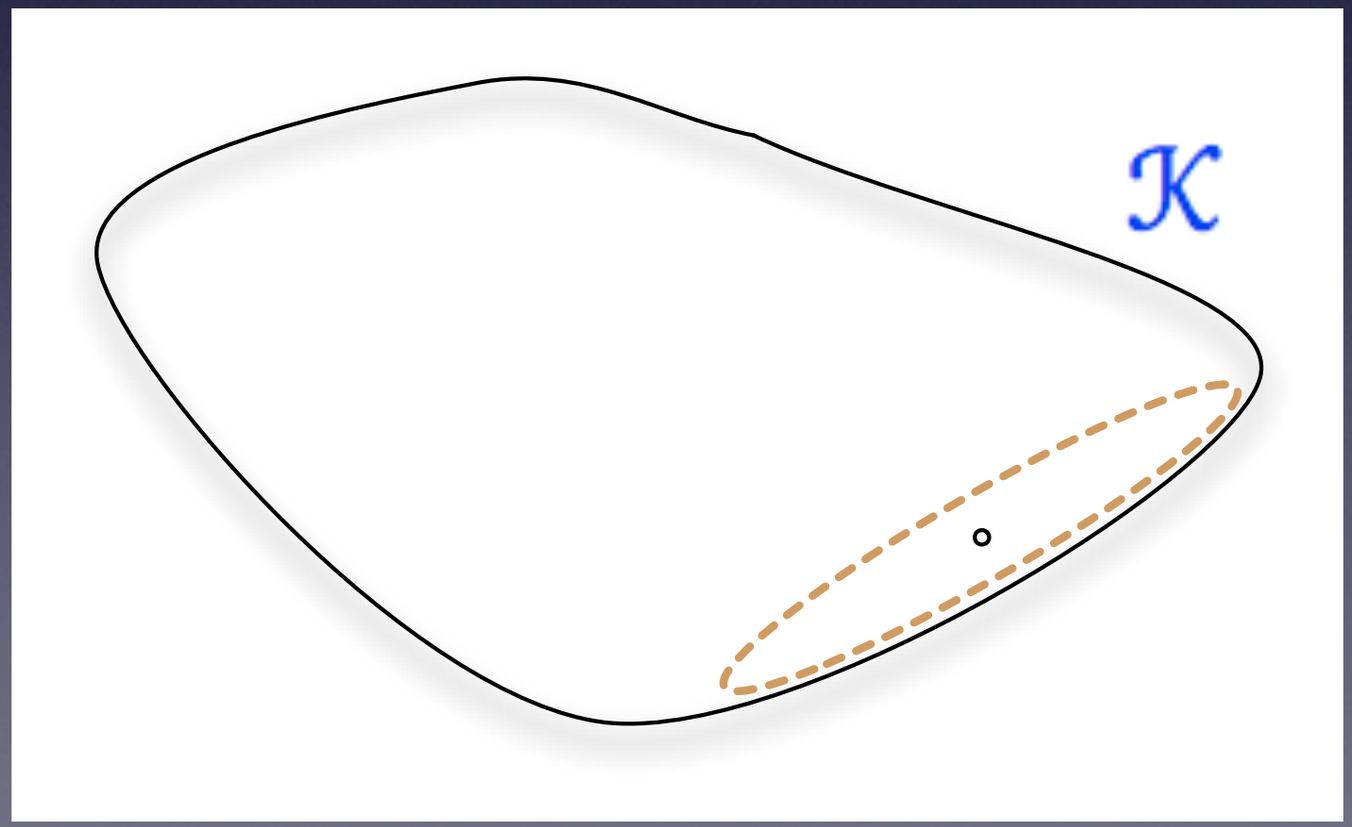
A Bonus Prize:
The Dikin Ellipsoid

$$\{\mathbf{z} \ : \ (\mathbf{z} - \mathbf{x}_t)\nabla^2\mathcal{R}(\mathbf{x}_t)(\mathbf{z} - \mathbf{x}_t) \leq 1\}$$



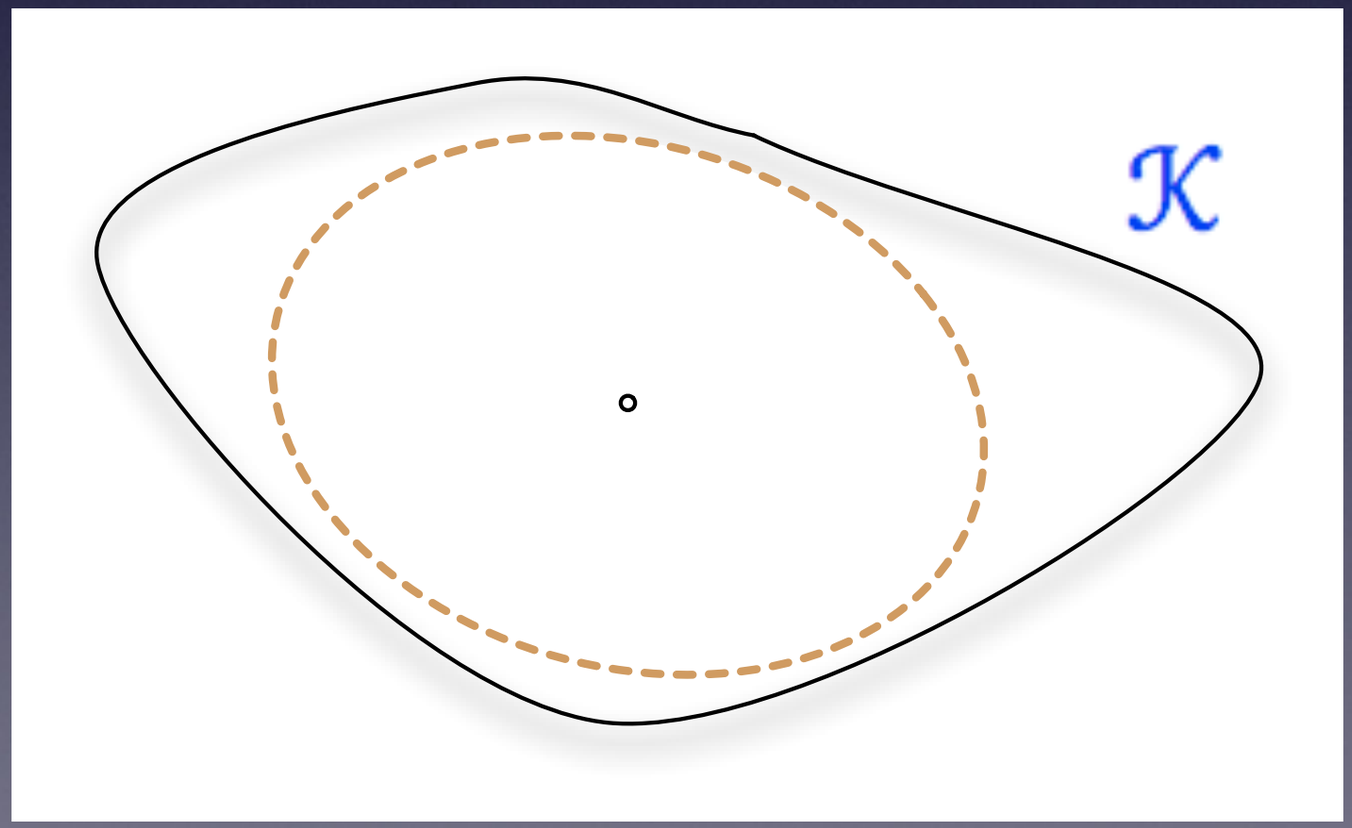
A Bonus Prize:
The Dikin Ellipsoid

$$\{\mathbf{z} \ : \ (\mathbf{z} - \mathbf{x}_t)\nabla^2\mathcal{R}(\mathbf{x}_t)(\mathbf{z} - \mathbf{x}_t) \leq 1\}$$



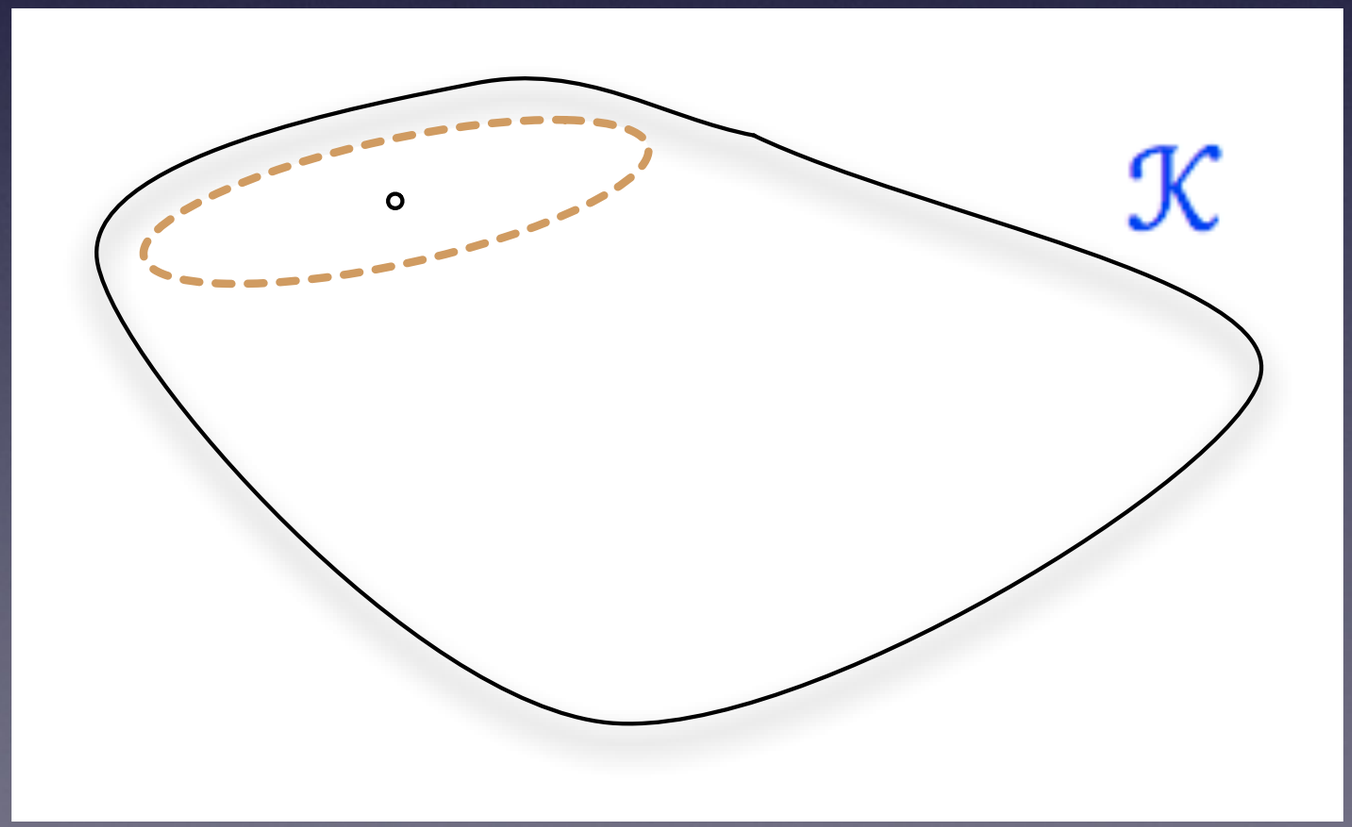
A Bonus Prize: The Dikin Ellipsoid

$$\{\mathbf{z} \ : \ (\mathbf{z} - \mathbf{x}_t)\nabla^2\mathcal{R}(\mathbf{x}_t)(\mathbf{z} - \mathbf{x}_t) \leq 1\}$$

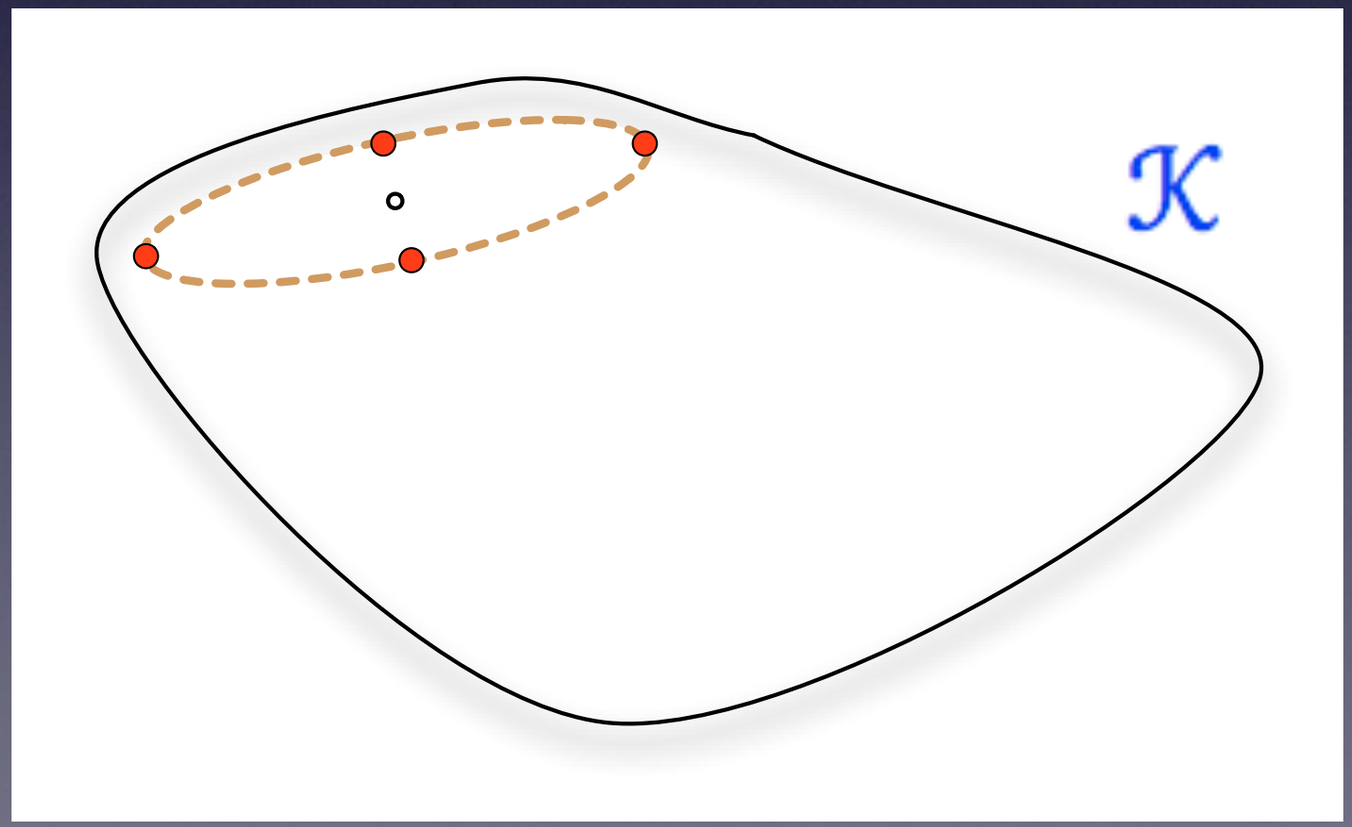


A Bonus Prize: The Dikin Ellipsoid

$$\{\mathbf{z} \ : \ (\mathbf{z} - \mathbf{x}_t)\nabla^2\mathcal{R}(\mathbf{x}_t)(\mathbf{z} - \mathbf{x}_t) \leq 1\}$$



A Bonus Prize:
The Dikin Ellipsoid



Newton's Method

Orig. Objective: $\min_{x \in D} g(x)$

Newton's Method

Orig. Objective: $\min_{x \in D} g(x)$

Regularized Obj.: $\min_{x \in D} \underbrace{g(x) + \lambda R(x)}_{\hat{g}(x)}$

Self-concord.

Newton's Method

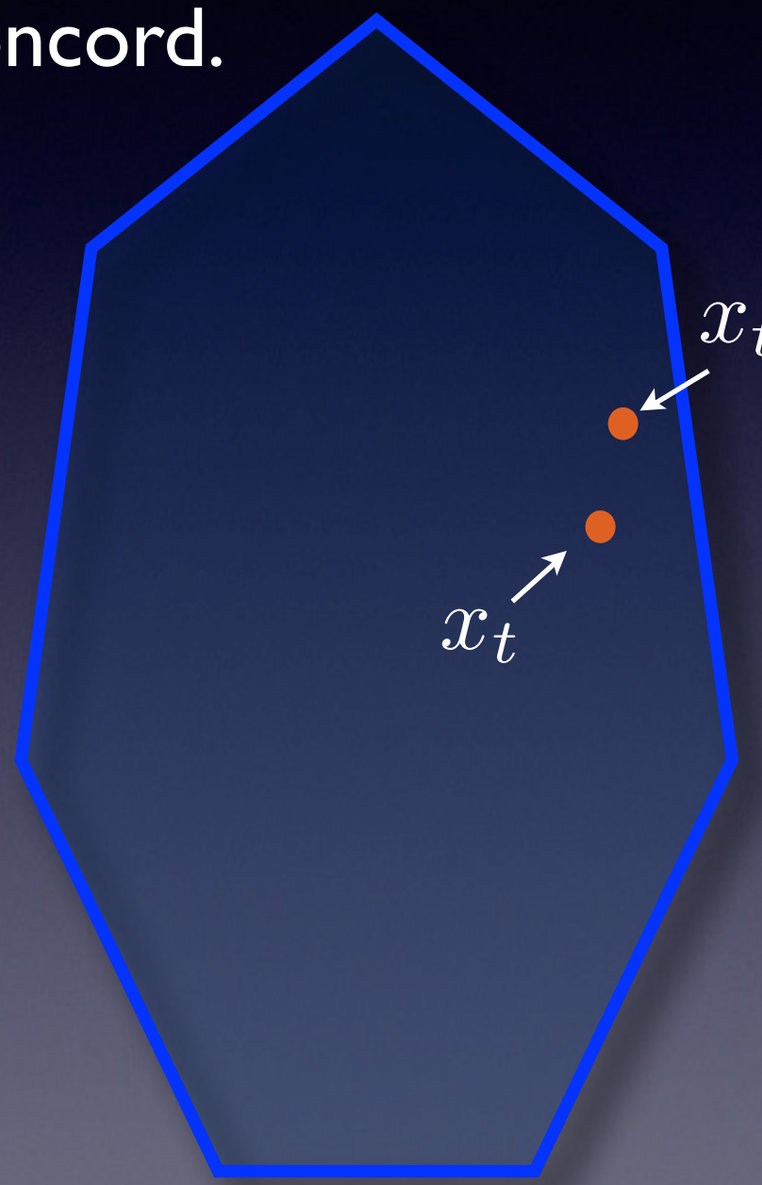
Orig. Objective: $\min_{x \in D} g(x)$

Regularized Obj.: $\min_{x \in D} \underbrace{g(x) + \lambda R(x)}_{\hat{g}(x)}$

Self-concord.

Newton Update:

$$x_{t+1} \leftarrow x_t + (\nabla^2_{x_t} R)^{-1} \nabla \hat{g}(x_t)$$



Newton's Method

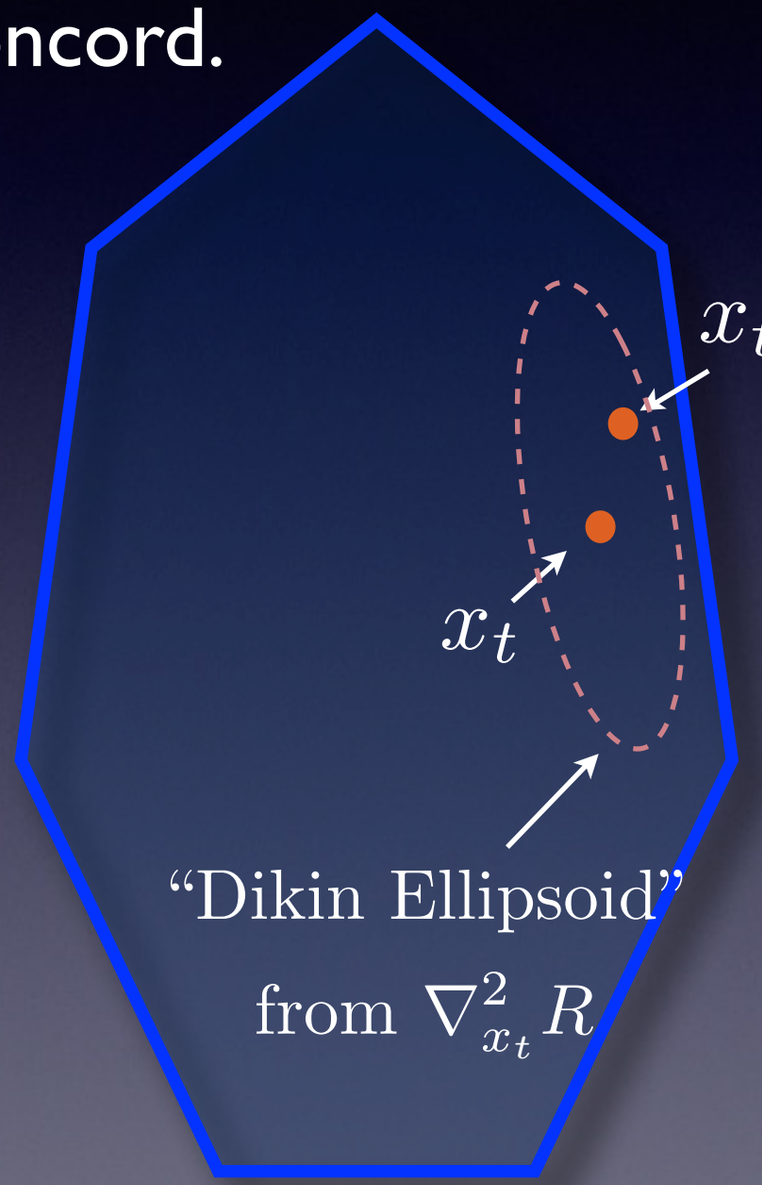
Orig. Objective: $\min_{x \in D} g(x)$

Regularized Obj.: $\min_{x \in D} \underbrace{g(x) + \lambda R(x)}_{\hat{g}(x)}$

Self-concord.

Newton Update:

$$x_{t+1} \leftarrow x_t + (\nabla^2_{x_t} R)^{-1} \nabla \hat{g}(x_t)$$



Newton's Method

Orig. Objective: $\min_{x \in D} g(x)$

Regularized Obj.: $\min_{x \in D} \underbrace{g(x) + \lambda R(x)}_{\hat{g}(x)}$

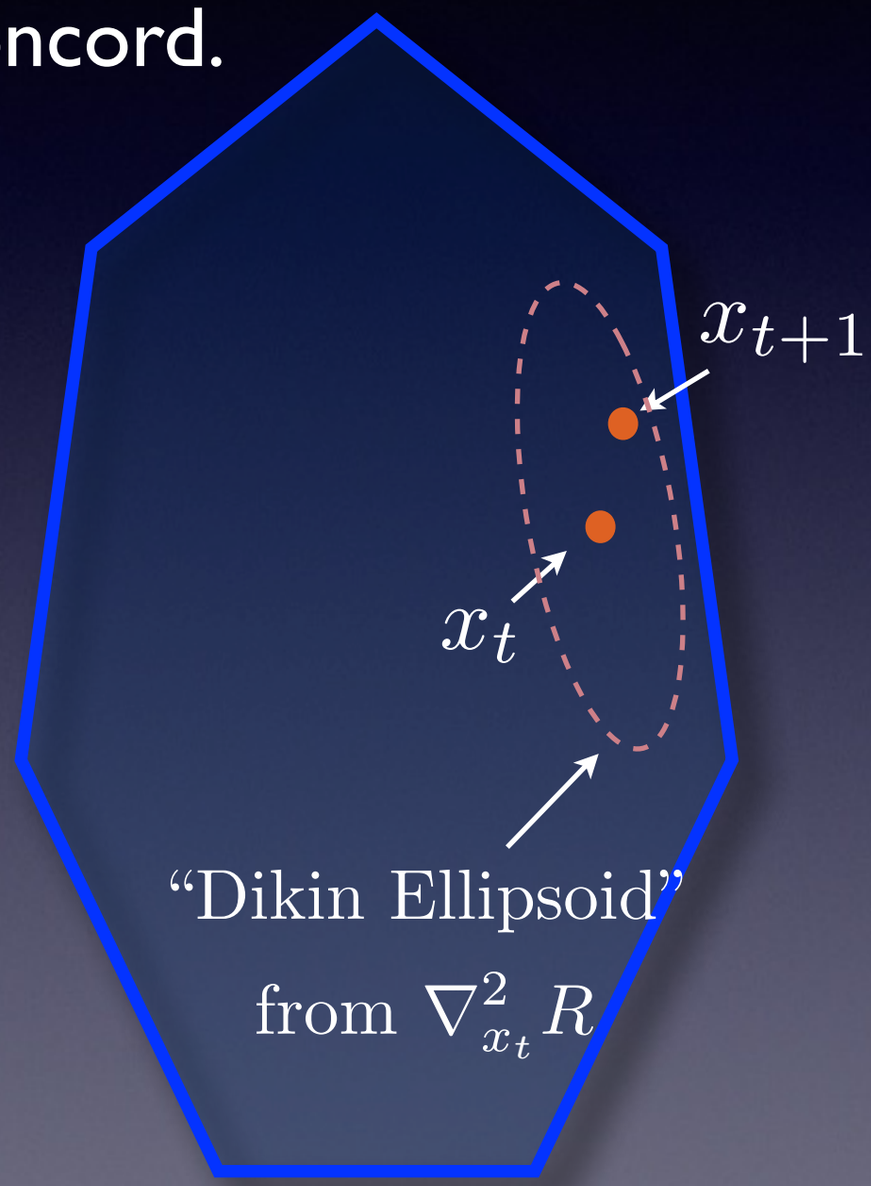
Self-concord.

Newton Update:

$$x_{t+1} \leftarrow x_t + (\nabla^2_{x_t} R)^{-1} \nabla \hat{g}(x_t)$$

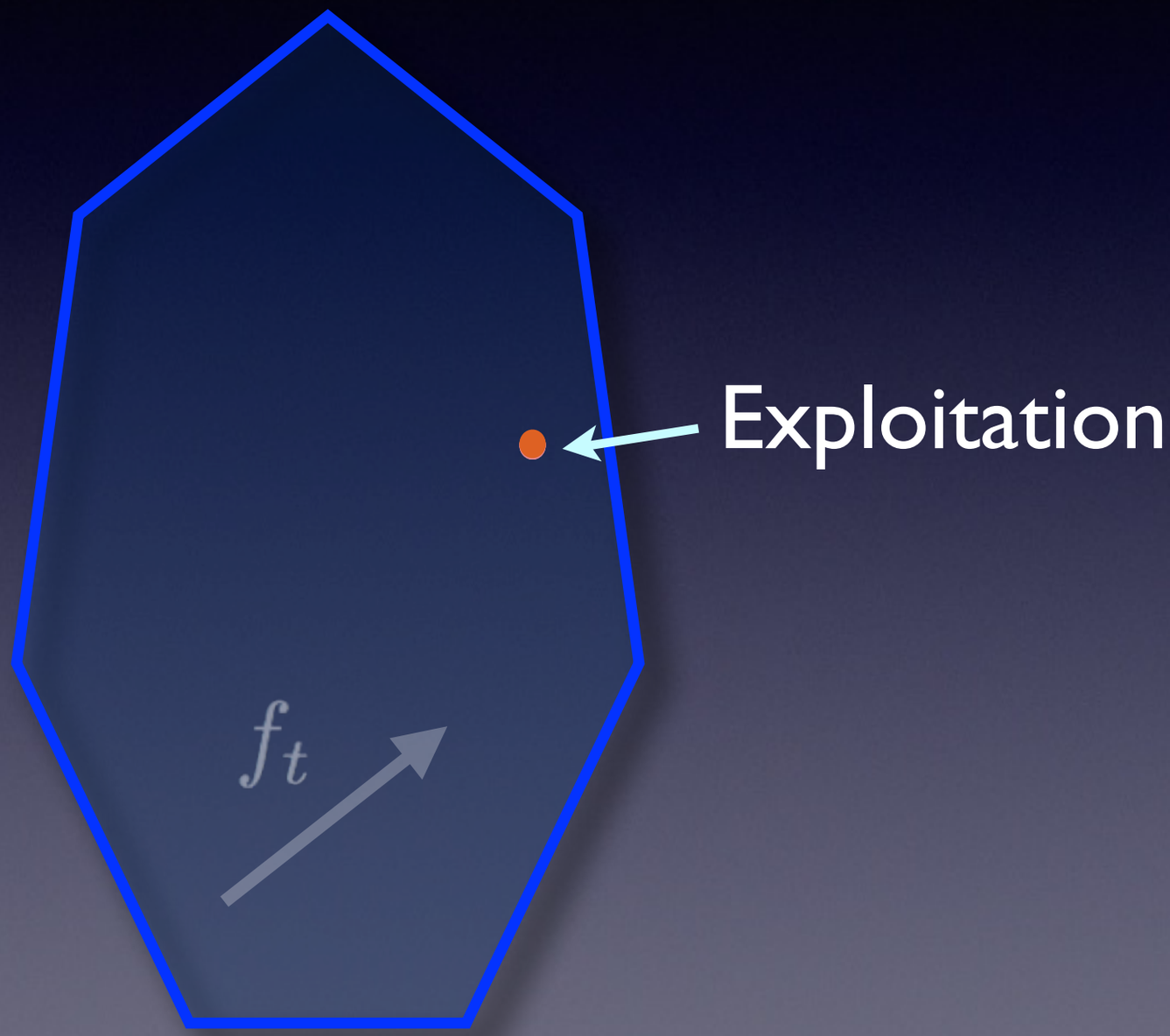
Important Fact:

$$x_{t+1} \in (\nabla^2_{x_t} R)\text{-ellipsoid} \subset D$$

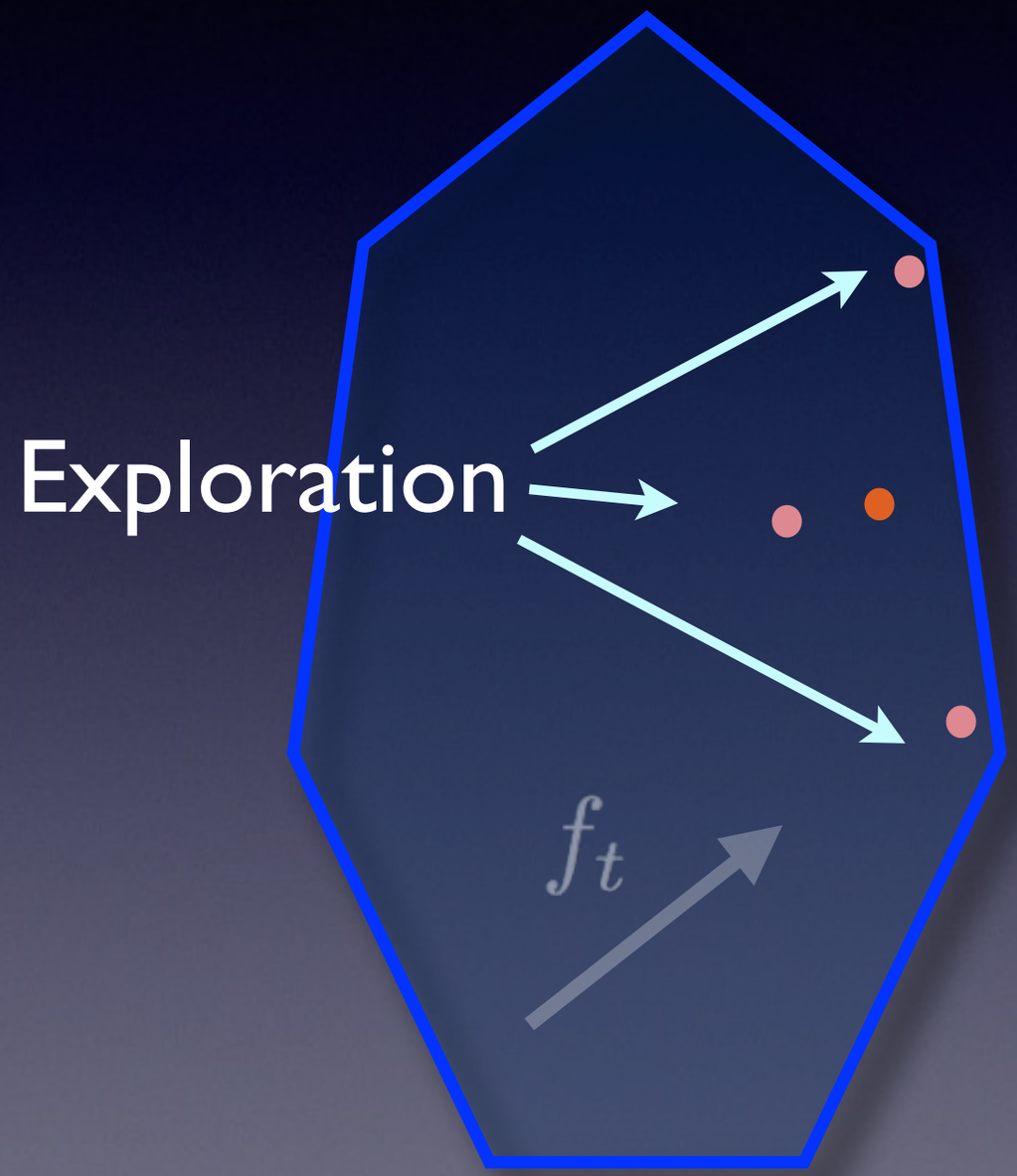


Back to Bandit Optimization: Estimation Troubles

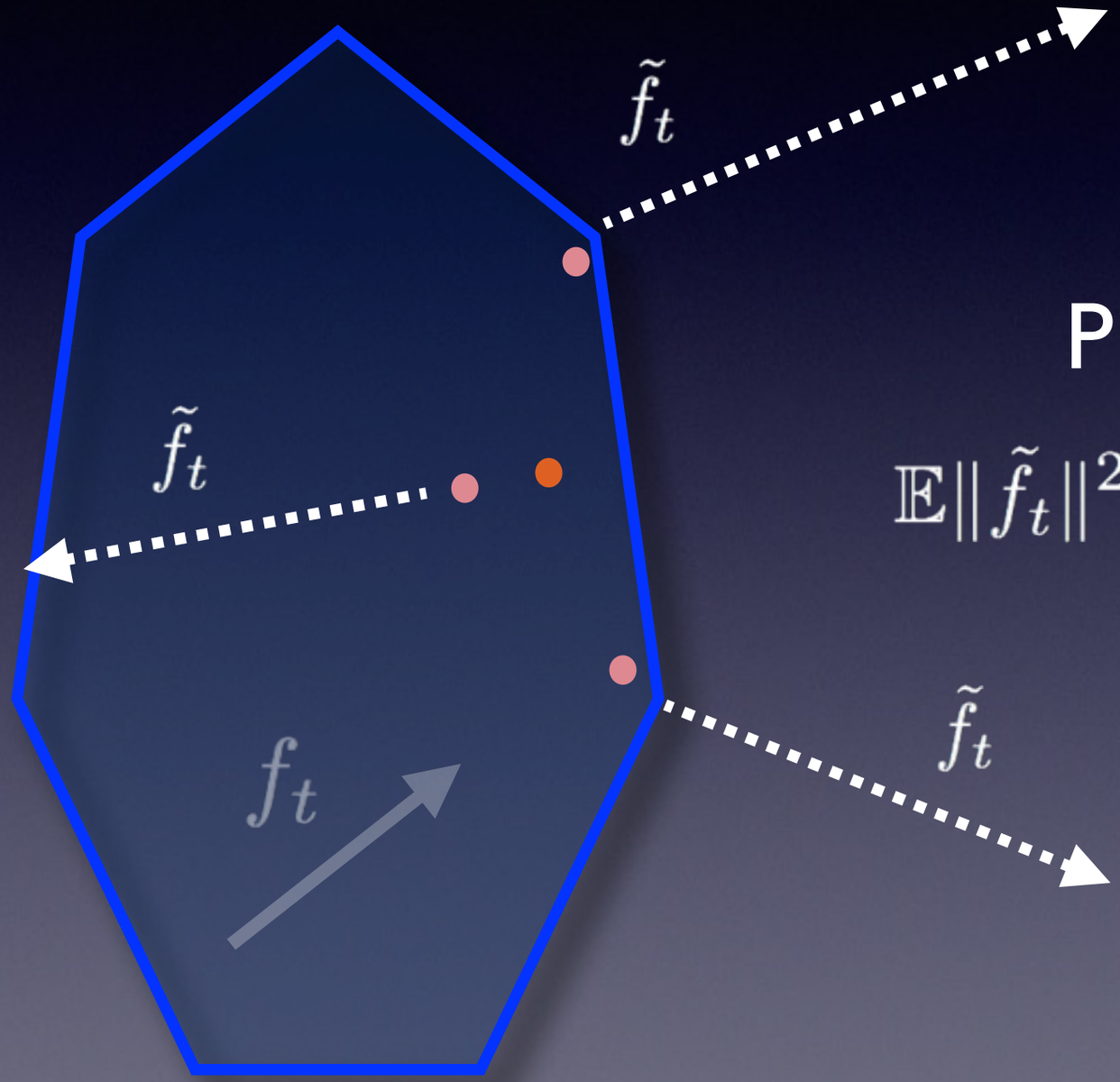
Back to Bandit Optimization: Estimation Troubles



Back to Bandit Optimization: Estimation Troubles



Back to Bandit Optimization: Estimation Troubles



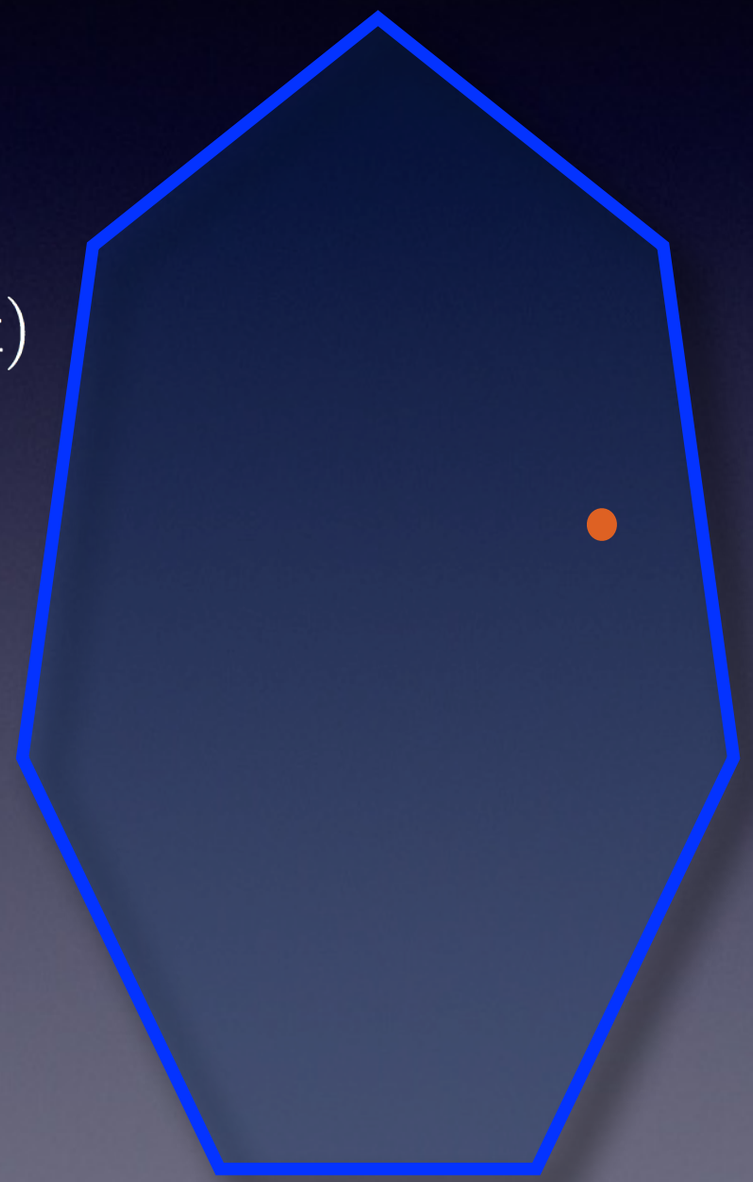
Problem:

$$\mathbb{E}\|\tilde{f}_t\|^2 \approx \frac{1}{\mathrm{dist}(x_t, \mathrm{bndry})}$$

Solution: Regularize w/ Self-concordant Function

R self-concordant barrier fun. on $\mathcal{K}$

Update: $\mathbf{x}_t := \arg\min_{\mathbf{x}\in\mathcal{K}} \sum_{s=1}^{t-1} \mathbf{f}_s \cdot \mathbf{x} + \lambda R(\mathbf{x})$



Solution: Regularize w/ Self-concordant Function

R self-concordant barrier fun. on $\mathcal{K}$

Update: $\mathbf{x}_t := \arg\min_{\mathbf{x}\in\mathcal{K}} \sum_{s=1}^{t-1} \mathbf{f}_s \cdot \mathbf{x} + \lambda R(\mathbf{x})$

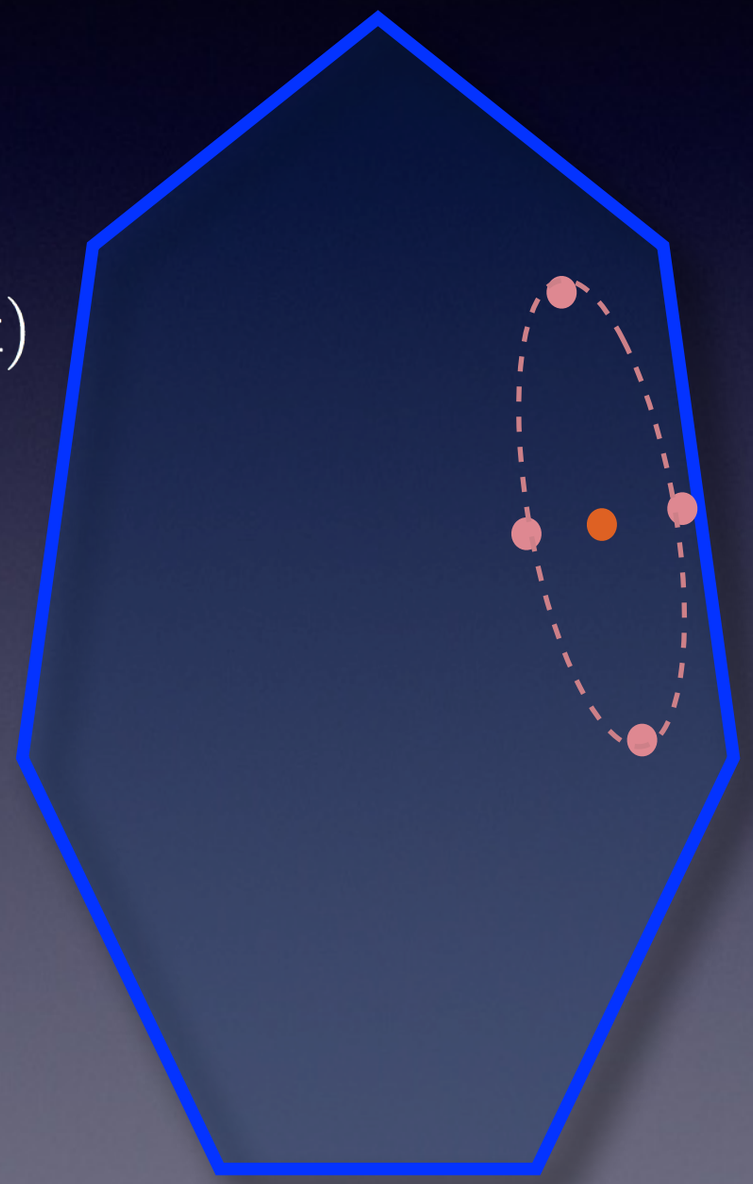
Solution: Regularize w/ Self-concordant Function

R self-concordant barrier fun. on $\mathcal{K}$

Update: $\mathbf{x}_t := \arg\min_{\mathbf{x}\in\mathcal{K}} \sum_{s=1}^{t-1} \mathbf{f}_s \cdot \mathbf{x} + \lambda R(\mathbf{x})$

Sample from:

eigenpoles of $(\nabla^2_{x_t} R)$ – ellipsoid



Solution: Regularize w/ Self-concordant Function

R self-concordant barrier fun. on $\mathcal{K}$

Update: $\mathbf{x}_t := \arg\min_{\mathbf{x}\in\mathcal{K}} \sum_{s=1}^{t-1} \mathbf{f}_s \cdot \mathbf{x} + \lambda R(\mathbf{x})$

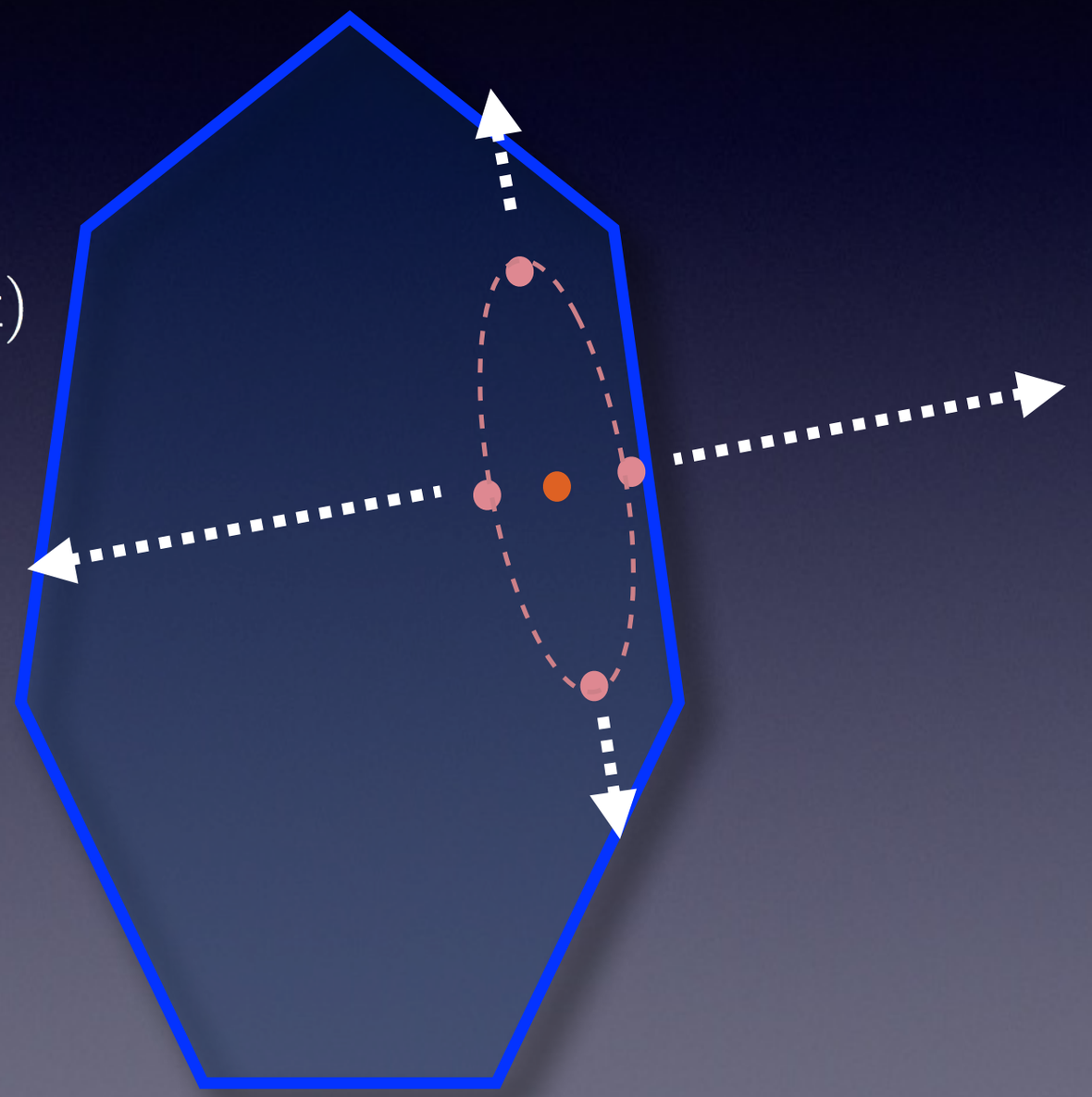
Sample from:

eigenpoles of $(\nabla^2_{x_t} R)$ – ellipsoid

Estimate sizes: $\tilde{f}_t \approx \lambda_i^{1/2} e_i$

$\{\lambda_j\}$ eigenvals of $\nabla^2_{x_t} R$

$\{e_j\}$ unit eigenvecs of $\nabla^2_{x_t} R$



FTRL Regret Bound

$$\text{Regret}_T \leq \sum_{t=1}^{T} \lambda D_R(\mathbf{x}_t, \mathbf{x}_{t+1}) + \lambda R(\mathbf{x}^*)$$

$$\leq \sum_{t=1}^{T} \frac{\|\tilde{\mathbf{f}}_t\|_*^2}{\lambda} + \lambda R(\mathbf{x}^*)$$

$$\leq \frac{T \cdot G}{\lambda} + \lambda D \leq 2\sqrt{T \cdot G \cdot D}$$

FTRL Regret Bound

$$\text{Regret}_T \leq \sum_{t=1}^{T} \lambda D_R(\mathbf{x}_t, \mathbf{x}_{t+1}) + \lambda R(\mathbf{x}^*)$$

Grows large!

$$\leq \sum_{t=1}^{T} \frac{\|\tilde{\mathbf{f}}_t\|_*^2}{\lambda} + \lambda R(\mathbf{x}^*)$$

$$\leq \frac{T \cdot G}{\lambda} + \lambda D \leq 2\sqrt{T \cdot G \cdot D}$$

FTRL Regret Bound

$$\text{Regret}_T \leq \sum_{t=1}^{T} \lambda D_R(\mathbf{x}_t, \mathbf{x}_{t+1}) + \lambda R(\mathbf{x}^*)$$

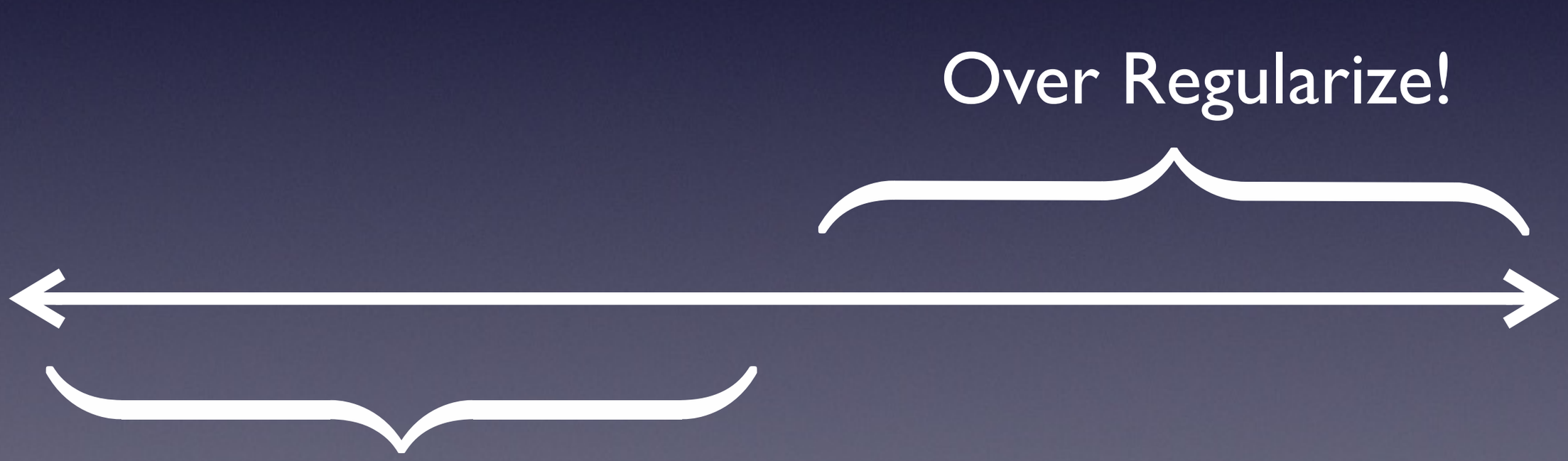
$$\leq \sum_{t=1}^{T} \frac{\tilde{\mathbf{f}}_t^\top (\nabla_{\mathbf{x}_t}^{-2} R)\, \tilde{\mathbf{f}}_t}{\lambda} + \lambda R(\mathbf{x}^*)$$

FTRL Regret Bound

$$\begin{aligned}
\mathrm{Regret}_T &\leq \sum_{t=1}^{T} \lambda D_R(\mathbf{x}_t, \mathbf{x}_{t+1}) + \lambda R(\mathbf{x}^*) \\
&\leq \sum_{t=1}^{T} \frac{\tilde{\mathbf{f}}_t^\top (\nabla_{\mathbf{x}_t}^{-2} R)\, \tilde{\mathbf{f}}_t}{\lambda} + \lambda R(\mathbf{x}^*) \\
&\leq \sum_{t=1}^{T} \frac{nG\sqrt{\sigma_{i_t}}\sigma_{i_t}^{-1}\sqrt{\sigma_{i_t}}}{\lambda} + \lambda D\theta \log T \\
&\leq 2\sqrt{n \cdot G \cdot D \cdot \theta \cdot T \log T}
\end{aligned}$$

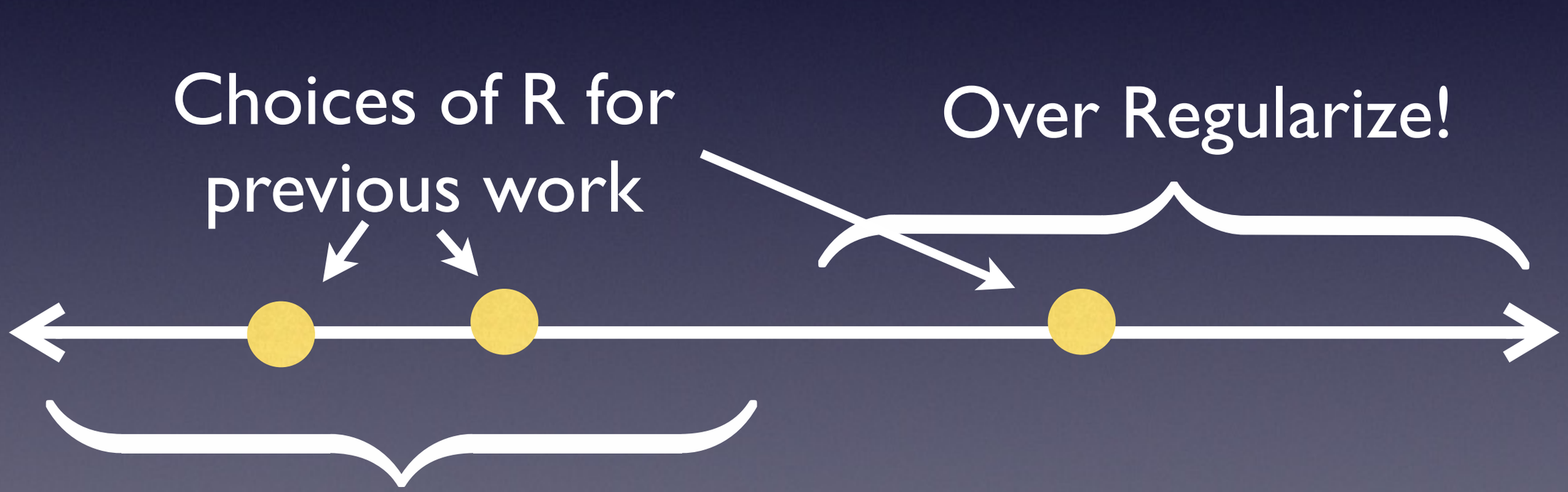
Little Flexibility in Selecting R

X-axis: how fast does $\|\nabla^2 R\|$ grow w.r.t. 1/dist-bndry?



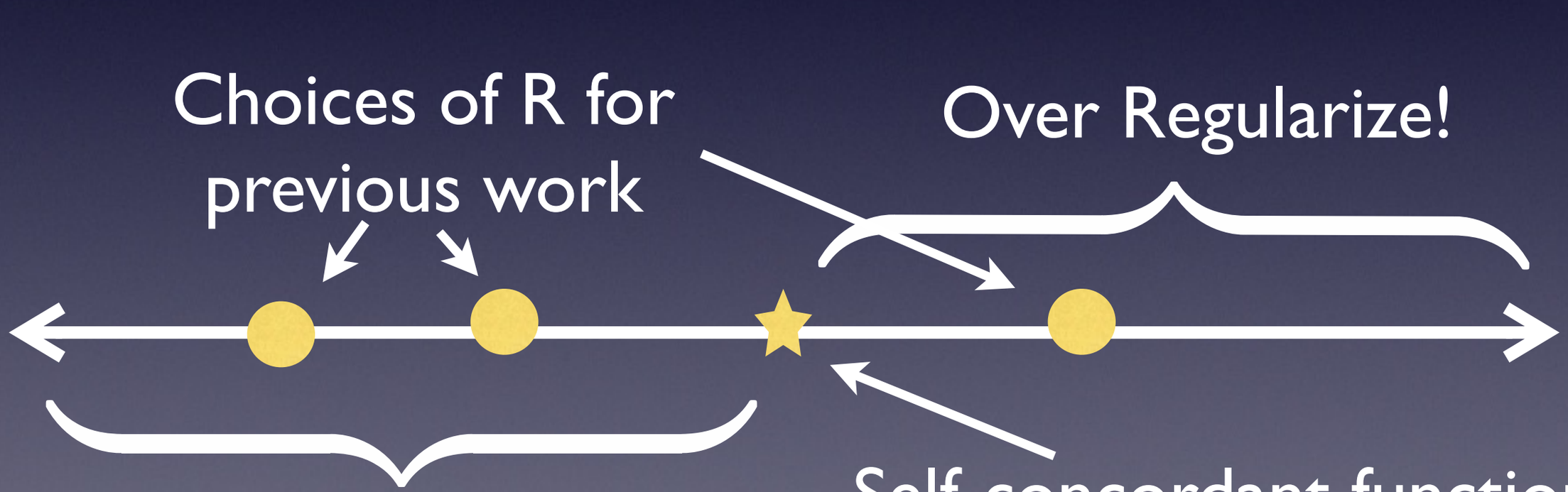
Little Flexibility in Selecting R

X-axis: how fast does $\|\nabla^2 R\|$ grow w.r.t. 1/dist-bndry?



Little Flexibility in Selecting R

X-axis: how fast does $\|\nabla^2 R\|$ grow w.r.t. 1/dist-bndry?



High Probability and Adaptive Adversaries

- These results are "in expectation" only! And only work against "oblivious adversaries"
- Achieving bounds with **high probability** and against **adaptive adversaries** is much more challenging
- We've made some progress on this (Abernethy/Rakhlin COLT 2009)
- Still, the general problem is hard...

Open Problem 1

- For the full-information setting, several "online learning games" have been fully solved. Minimax solution is very nice.
- Can the same be achieved for bandit setting?

Open Problem 2

- Another problem, which *looks* like Online Convex Opt, is the bandit multiclass setting
- On each round, a Learner sees an example and predicts a class. Told "right" or "wrong"
- Current best algorithms get $O(T^{2/3})$ regret
- Should be an efficient method for $O(T^{1/2})$

Thanks!