Contextual Inverse Optimization: Offline and Online Learning

Omar Besbes, Yuri Fonseca and Ilan Lobel

Columbia, Columbia and NYU

Simons Institute, September 2022

Contextual Inverse Optimization

- Standard data-driven decision processes framework:
  - Given context, choose action, observe reward.
- In many settings, **rewards cannot be observed**.
  - Is there other type of feedback that we can use to learn?

Contextual Inverse Optimization

- Standard data-driven decision processes framework:
  - Given context, choose action, observe reward.
- In many settings, **rewards cannot be observed**.
  - Is there other type of feedback that we can use to learn?
- In this work we consider problems where the reward is not observed but we observe, after-the-fact, what you **should** have done.
  - Contextual inverse optimization

Contextual Inverse Optimization

- Standard data-driven decision processes framework:
  - Given context, choose action, observe reward.
- In many settings, **rewards cannot be observed**.
  - Is there other type of feedback that we can use to learn?
- In this work we consider problems where the reward is not observed but we observe, after-the-fact, what you **should** have done.
  - Contextual inverse optimization
- Applications:
  - Economics: learn from revealed preferences.
  - Robotics: teach a robot or AV by demonstration.
  - Medicine: learn from a doctor's decision-making.

Problem Formulation

In every t, you would like to solve:

$$\min_{x \in \mathcal{X}_t} f_t(x)' c^\star$$

We **don't know** $c^\star$, but we observe $\mathcal{X}_t$, $f_t(\cdot)$ and $x_t^\star$ (after period t):

$$x_t^\star \in \arg\min_{x \in \mathcal{X}_t} f_t(x)' c^\star$$

Problem Formulation

In every t, you would like to solve:

$$\min_{x \in \mathcal{X}_t} f_t(x)' c^\star$$

We **don't know** $c^\star$, but we observe $\mathcal{X}_t$, $f_t(\cdot)$ and $x_t^\star$ (after period t):

$$x_t^\star \in \arg\min_{x \in \mathcal{X}_t} f_t(x)' c^\star$$

Example: Learning from Revealed Preferences

$$x_t^\star = \arg\max_{x \in \mathcal{X}_t} x' Z_t c^\star \ : \ x' p_t \leq b_t, \quad \mathcal{X}_t = \{0,1\}^{n_t}$$

Related Literature: Inverse Optimization

Estimate cost vector based on optimal action

- Ajuha and Orlin (OR 2001)

What if you have many data points?

- Esfahani, Shafieezadeh-Abadeh, Hanasuanto and Kuhn (MP 2018): closest to our offline model, stochastic framework
- Bärmann, Pokutta and Schneider (ICML 2017): closest to our online model, gradient descent approach

Related Literature: Contextual Pricing and Search

Class of contextual bandit models where nature picks context adversarially and we choose action.

- Cohen, Lobel and Paes Leme (MS 2020): ellipsoid method
- Lobel, Paes Leme and Vladu (OR 2018): centroid and projection
- Paes Leme and Schneider (FOCS 2018): intrinstic volume
- Krishnamurthy, Lykouris, Podimata and Schapire (STOC 2021): irrational agents

We leverage ideas from this literature, but the problems are of a different nature (we have far less control on the feedback).

Related Literature: Structured Prediction and Inverse Reinforcement Learning

Optimization-based structured prediction is similar to inverse optimization but focuses on a different metric (prediction error).

- Taskar, Chatalbashev, Koller and Guestrin (ICML 2005): SVM-style approach called maximum margin planning
- Ratliff, Bagnell and Zinkevich (ICML 2006): online version

If you assign linear functionals to features, this approach can be used to learn a reward function in reinforcement learning.

- Abbeel and Ng (ICML 2004): apprenticeship learning

Main Results

Offline setting:

- We propose a geometric definition of data informativeness.
- Using this notion, we characterize the minimax regret.

Main Results

Offline setting:

- We propose a geometric definition of data informativeness.
- Using this notion, we characterize the minimax regret.

Online setting:

- State-of-the-art: Bärmann et al. (ICML 2017) obtain $O(\sqrt{T})$ regret, assuming linear context functions.
- We obtain $O(d^4 \ln T)$ regret, assuming Lipschitz context functions.

Offline Setting: The Data

In the offline setting, we have N observations, and for $i = 1, ..., N$, we have:

- A set of feasible actions $\mathcal{X}_i \subset \mathbb{R}^n$
- A context function $f_i : \mathcal{X}_i \to \mathbb{R}^d$
- An optimal action $x_i^\star \in \mathcal{X}_i$

$$x_i^\star \in \arg\min_{x \in \mathcal{X}_i} f_i(x)' c^\star \quad \text{for some unknown } c^\star$$

Offline Setting: The Data

In the offline setting, we have N observations, and for $i = 1, ..., N$, we have:

- A set of feasible actions $\mathcal{X}_i \subset \mathbb{R}^n$
- A context function $f_i : \mathcal{X}_i \to \mathbb{R}^d$
- An optimal action $x_i^\star \in \mathcal{X}_i$

$$x_i^\star \in \underset{x \in \mathcal{X}_i}{\arg\min}\, f_i(x)' c^\star \quad \text{for some unknown } c^\star$$

Given the data $\mathcal{D} = (\mathcal{X}_i, f_i, x_i^\star)_{i=1,...,N}$ and initial knowledge set $c^\star \in C_0$, the set of feasible cost vectors is:

$$C(\mathcal{D}) = \left\{ c \in C_0 : x_i^\star \in \underset{x \in \mathcal{X}_i}{\arg\min}\, f_i(x)' c,\ \ i = 1, ..., N \right\}$$

Policy and Objective

A policy $\pi \in \mathcal{P}$ is a mapping from $(\mathcal{D}, \mathcal{X}, f)$ to an action $x^\pi \in \mathcal{X}$

Our **regret** is given by:

$$\mathcal{R}^\pi(c^\star, \mathcal{X}, f) = f(x^\pi)' c^\star - f(x^\star)' c^\star$$

Our **objective** is to find $\pi \in \mathcal{P}$ that minimizes the worst-case regret:

$$\mathrm{WCR}^\pi(\mathcal{D}) = \sup_{c^\star \in C(\mathcal{D}),\ \mathcal{X} \in \mathcal{B},\ f \in \mathcal{F}} \mathcal{R}^\pi(c^\star, \mathcal{X}, f)$$

- $\mathcal{B}$: set of all measurable subsets of $\mathbb{R}^n$ with diameter at most 1
- $\mathcal{F}$: set of all 1-Lipschitz continuous functions from $\mathbb{R}^n$ to $\mathbb{R}^d$

Offline Learning in an Adversarial Setting

Without distributional assumptions, we can't make claims about the convergence of the minimax regret as N grows.

- In a worst-case scenario ($\mathcal{X}_i$ and f_i are identical for all $i = 1, ..., N$), we wouldn't learn anything from observations $2, ..., N$.

Offline Learning in an Adversarial Setting

Without distributional assumptions, we can't make claims about the convergence of the minimax regret as N grows.

- In a worst-case scenario ($\mathcal{X}_i$ and f_i are identical for all $i = 1, ..., N$), we wouldn't learn anything from observations $2, ..., N$.

We need a bound that is strong **if the data is informative**.

Offline Learning in an Adversarial Setting

Without distributional assumptions, we can't make claims about the convergence of the minimax regret as N grows.

- In a worst-case scenario ($\mathcal{X}_i$ and f_i are identical for all $i = 1, ..., N$), we wouldn't learn anything from observations $2, ..., N$.

We need a bound that is strong **if the data is informative**.

- What does it mean for the data to be informative?

Offline Learning in an Adversarial Setting

Without distributional assumptions, we can't make claims about the convergence of the minimax regret as N grows.

- In a worst-case scenario ($\mathcal{X}_i$ and f_i are identical for all $i = 1, \ldots, N$), we wouldn't learn anything from observations $2, \ldots, N$.

We need a bound that is strong **if the data is informative**.

- What does it mean for the data to be informative?

We will build a **geometric** notion of what is an **informative** data set $\mathcal{D}$.

Proxy Policies

We focus our attention on **proxy policies**, which are policies where we pick action x^π according to a **proxy cost vector** c^π:

$$\mathcal{P}' = \left\{ \pi \in \mathcal{P} : \; x^\pi \in \arg\min_{x \in \mathcal{X}} f(x)' c^\pi, \; \text{for some } c^\pi \in \mathbb{S}^d \right\}$$

Proxy Policies

We focus our attention on **proxy policies**, which are policies where we pick action x^π according to a **proxy cost vector** c^π:

$$\mathcal{P}' = \left\{\pi \in \mathcal{P} : \; x^\pi \in \arg\min_{x \in \mathcal{X}} f(x)' c^\pi, \; \text{for some } c^\pi \in \mathbb{S}^d\right\}$$

Given a proxy cost c^π and a true cost $c^\star$, our loss is bounded by:

$$\mathcal{L}(c^\pi, c^\star) = \sup\left\{ f(x^\pi)' c^\star - f(x^\star)' c^\star : x^\pi \in \arg\min_{x \in \mathcal{X}} f(x)' c^\pi, \mathcal{X} \in \mathcal{B}, f \in \mathcal{F} \right\}$$

Proxy Policies

We focus our attention on **proxy policies**, which are policies where we pick action x^π according to a **proxy cost vector** c^π:

$$\mathcal{P}' = \left\{ \pi \in \mathcal{P} : \; x^\pi \in \arg\min_{x \in \mathcal{X}} f(x)' c^\pi, \; \text{for some } c^\pi \in \mathbb{S}^d \right\}$$

Given a proxy cost c^π and a true cost $c^\star$, our loss is bounded by:

$$\mathcal{L}(c^\pi, c^\star) = \sup \left\{ f(x^\pi)' c^\star - f(x^\star)' c^\star : x^\pi \in \arg\min_{x \in \mathcal{X}} f(x)' c^\pi, \mathcal{X} \in \mathcal{B}, f \in \mathcal{F} \right\}$$

For any data set $\mathcal{D}$:

$$\inf_{\pi \in \mathcal{P}'} \mathrm{WCR}^\pi(\mathcal{D}) = \inf_{c^\pi \in \mathbb{S}^d} \sup_{c^\star \in C(\mathcal{D})} \mathcal{L}(c^\pi, c^\star)$$

The Loss of a Proxy Policy

Lemma

Let θ be the angle between two vectors. For any c^π, $c^\star \in \mathbb{S}^d$:

$$\mathcal{L}(c^\pi, c^\star) = \begin{cases} \sin\theta(c^\pi, c^\star) & \textit{if } \theta(c^\pi, c^\star) \leq \pi/2 \\ 1 & \textit{otherwise} \end{cases}$$

The Loss of a Proxy Policy

Lemma

Let θ be the angle between two vectors. For any c^π, $c^\star \in \mathbb{S}^d$:

$$\mathcal{L}(c^\pi, c^\star) = \begin{cases} \sin\theta(c^\pi, c^\star) & \textit{if } \theta(c^\pi, c^\star) \leq \pi/2 \\ 1 & \textit{otherwise} \end{cases}$$

If the angle between the true cost $c^\star$ and the proxy cost c^π is small, the regret must also be small.

The Loss of a Proxy Policy

Lemma

Let θ be the angle between two vectors. For any c^{π}, $c^{\star} \in \mathbb{S}^{d}$:

$$\mathcal{L}(c^{\pi}, c^{\star}) = \begin{cases} \sin\theta(c^{\pi}, c^{\star}) & \textit{if } \theta(c^{\pi}, c^{\star}) \leq \pi/2 \\ 1 & \textit{otherwise} \end{cases}$$

If the angle between the true cost $c^{\star}$ and the proxy cost c^{π} is small, the regret must also be small.

We prove this lemma by showing that the problem of finding a worst-case loss is a semi-definite program.

Uncertainty Angle and Circumcenter

Definition

We define the **uncertainty angle** of a set C to be:

$$\alpha(C) = \inf_{\hat{c} \in \mathbb{S}^d} \sup_{c^\star \in C} \theta(\hat{c}, c^\star),$$

Uncertainty Angle and Circumcenter

Definition

We define the **uncertainty angle** of a set C to be:

$$\alpha(C) = \inf_{\hat{c} \in \mathbb{S}^d} \sup_{c^\star \in C} \theta(\hat{c}, c^\star),$$

Lemma

The minimizer $\hat{c}$ exists and we call it the **circumcenter** *of C.*

Uncertainty Angle and Circumcenter

Definition

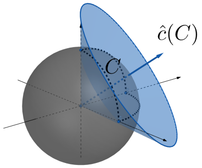
We define the **uncertainty angle** of a set C to be:

$$\alpha(C) = \inf_{\hat{c} \in \mathbb{S}^d} \sup_{c^\star \in C} \theta(\hat{c}, c^\star),$$

Lemma

The minimizer $\hat{c}$ exists and we call it the **circumcenter** *of C.*

The uncertainty angle and the circumcenter are the aperture and the axis of the smallest revolution cone containing C.



The Circumcenter Policy

Definition

We call the proxy policy that uses the circumcenter as the proxy cost the **circumcenter policy**.

The Circumcenter Policy

Definition

We call the proxy policy that uses the circumcenter as the proxy cost the **circumcenter policy**.

Theorem

The optimal proxy policy is the circumcenter policy. It achieves:

$$\inf_{\pi \in \mathcal{P}'} \mathrm{WCR}^{\pi}(\mathcal{D}) = \begin{cases} \sin \alpha(C(\mathcal{D})) & \textit{if } \alpha(C(\mathcal{D})) \leq \pi/2 \\ 1 & \textit{otherwise} \end{cases}$$

The Circumcenter Policy

Definition

We call the proxy policy that uses the circumcenter as the proxy cost the **circumcenter policy**.

Theorem

The optimal proxy policy is the circumcenter policy. It achieves:

$$\inf_{\pi \in \mathcal{P}'} \mathrm{WCR}^{\pi}(\mathcal{D}) = \begin{cases} \sin \alpha(C(\mathcal{D})) & \text{if } \alpha(C(\mathcal{D})) \leq \pi/2 \\ 1 & \text{otherwise} \end{cases}$$

- The **uncertainty angle** determines the worst-case regret.

The Circumcenter Policy

Definition

We call the proxy policy that uses the circumcenter as the proxy cost the **circumcenter policy**.

Theorem

The optimal proxy policy is the circumcenter policy. It achieves:

$$\inf_{\pi \in \mathcal{P}'} \mathrm{WCR}^{\pi}(\mathcal{D}) = \begin{cases} \sin \alpha(C(\mathcal{D})) & \text{if } \alpha(C(\mathcal{D})) \leq \pi/2 \\ 1 & \text{otherwise} \end{cases}$$

- The **uncertainty angle** determines the worst-case regret.
- Nontrivial bounds iff $\mathcal{D}$ implies feasible costs live in a **pointed cone**.

Online Setting: The Data

In the online setting, at each period $t = 1, ..., T$, we are given:

- A set of feasible actions $\mathcal{X}_t \subset \mathbb{R}^n$
- A context function $f_t : \mathcal{X}_t \to \mathbb{R}^d$

Online Setting: The Data

In the online setting, at each period $t = 1, ..., T$, we are given:

- A set of feasible actions $\mathcal{X}_t \subset \mathbb{R}^n$
- A context function $f_t : \mathcal{X}_t \to \mathbb{R}^d$

We then choose an action $x_t^\pi \in \mathcal{X}_t$

Online Setting: The Data

In the online setting, at each period $t = 1, ..., T$, we are given:

- A set of feasible actions $\mathcal{X}_t \subset \mathbb{R}^n$
- A context function $f_t : \mathcal{X}_t \to \mathbb{R}^d$

We then choose an action $x_t^\pi \in \mathcal{X}_t$

At the end of period t, we observe an optimal action $x_t^\star \in \mathcal{X}_t$

Online Setting: The Data

In the online setting, at each period $t = 1, \dots, T$, we are given:

- A set of feasible actions $\mathcal{X}_t \subset \mathbb{R}^n$
- A context function $f_t : \mathcal{X}_t \to \mathbb{R}^d$

We then choose an action $x_t^\pi \in \mathcal{X}_t$

At the end of period t, we observe an optimal action $x_t^\star \in \mathcal{X}_t$

Our data at the start of period t is given by

$$\mathcal{I}_t = (\mathcal{X}_i, f_i, x_i^\star, x_i^\pi)_{i=1,\dots,t-1} \cup (\mathcal{X}_t, f_t)$$

Online Setting: The Data

In the online setting, at each period $t = 1, ..., T$, we are given:

- A set of feasible actions $\mathcal{X}_t \subset \mathbb{R}^n$
- A context function $f_t : \mathcal{X}_t \to \mathbb{R}^d$

We then choose an action $x_t^\pi \in \mathcal{X}_t$

At the end of period t, we observe an optimal action $x_t^\star \in \mathcal{X}_t$

Our data at the start of period t is given by

$$\mathcal{I}_t = (\mathcal{X}_i, f_i, x_i^\star, x_i^\pi)_{i=1,...,t-1} \cup (\mathcal{X}_t, f_t)$$

The set of cost vectors compatible with our data at period t is:

$$C(\mathcal{I}_t) = \left\{ c \in C_0 : x_i^\star \in \arg\min_{x \in \mathcal{X}_i} f_i(x)' c, \; i = 1, ..., t-1 \right\}$$

Policy and Objective

A policy $\pi \in \mathcal{P}$ is a mapping from $\mathcal{I}_t$ to an action $x_t^\pi \in \mathcal{X}_t$

Our **cumulative regret** is given by:

$$\mathcal{R}_T^\pi\left(c^\star, \vec{\mathcal{X}}, \vec{f}\right) = \sum_{t=1}^{T}\Big(f_t(x_t^\pi)'c^\star - f_t(x_t^\star)'c^\star\Big)$$

Our **objective** is to find $\pi \in \mathcal{P}$ that minimizes the worst-case regret:

$$\mathrm{WCR}^\pi(C_0) \; = \sup_{c^\star \in C_0,\; \vec{\mathcal{X}} \in \mathcal{B}^T,\; \vec{f} \in \mathcal{F}^T} \mathcal{R}^\pi\left(c^\star, \vec{\mathcal{X}}, \vec{f}\right)$$

The Decision-Maker Does Not Control the Feedback

In related problems (contextual pricing and contextual search), the decision-maker has some control over the feedback it gets.

The Decision-Maker Does Not Control the Feedback

In related problems (contextual pricing and contextual search), the decision-maker has some control over the feedback it gets.

In our problem, the decision-maker has no direct control over the feedback.

The Decision-Maker Does Not Control the Feedback

In related problems (contextual pricing and contextual search), the decision-maker has some control over the feedback it gets.

In our problem, the decision-maker has no direct control over the feedback.

- The actions $\{x_t^\pi\}$ do not appear in the information set:

$$c^\star \in C(\mathcal{I}_t) = \left\{ c \in C_0 : x_i^\star \in \arg\min_{x \in \mathcal{X}_i} f_i(x)' c, \; i = 1, \ldots, t-1 \right\}$$

The Decision-Maker Does Not Control the Feedback

In related problems (contextual pricing and contextual search), the decision-maker has some control over the feedback it gets.

In our problem, the decision-maker has no direct control over the feedback.

- The actions $\{x_t^\pi\}$ do not appear in the information set:

$$c^\star \in C(\mathcal{I}_t) = \left\{ c \in C_0 : x_i^\star \in \arg\min_{x \in \mathcal{X}_i} f_i(x)'c, \;\; i = 1, \ldots, t-1 \right\}$$

- Perhaps we should ignore the dynamics and use a greedy policy.

The Decision-Maker Does Not Control the Feedback

In related problems (contextual pricing and contextual search), the decision-maker has some control over the feedback it gets.

In our problem, the decision-maker has no direct control over the feedback.

- The actions $\{x_t^\pi\}$ do not appear in the information set:

$$c^\star \in C(\mathcal{I}_t) = \left\{ c \in C_0 : x_i^\star \in \arg\min_{x \in \mathcal{X}_i} f_i(x)' c, \; i = 1, ..., t-1 \right\}$$

- Perhaps we should ignore the dynamics and use a greedy policy.
- Greedy = circumcenter policy.

The Circumcenter Policy Fails in the Online Setting

Theorem

There exists a C_0 such that, if the decision-maker uses the circumcenter policy, nature can cause regret that is linear in T.

The Circumcenter Policy Fails in the Online Setting

Theorem

There exists a C_0 such that, if the decision-maker uses the circumcenter policy, nature can cause regret that is linear in T.

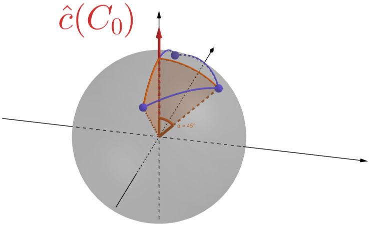
- Nature can construct instances where there the decision-maker simultaneously incurs large regret and learns essentially nothing.

The Circumcenter Policy Fails in the Online Setting

Theorem

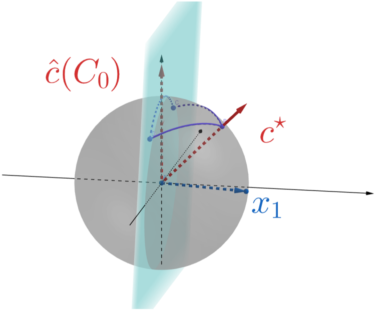
There exists a C_0 such that, if the decision-maker uses the circumcenter policy, nature can cause regret that is linear in T.

- Nature can construct instances where there the decision-maker simultaneously incurs large regret and learns essentially nothing.



Learning Nothing While Incurring Regret

- Feasible actions $\mathcal{X} = \{0, x_1\}$ and context function $f(x) = x$
- Proxy cost of $\hat{c}(C_0)$ implies x_1 is better
- With true cost $c^\star$, the actual optimal action is $x^\star = 0$
- Regret is substantial: ${x_1}'c^\star$
- Feedback is marginal: ${x_1}'c^\star \geq {x^\star}'c^\star = 0$



What Control Do We Have?

Let us assume $f_t(x_t^\star) \neq f_t(x_t^\pi)$ (otherwise we don't incur regret)

What Control Do We Have?

Let us assume $f_t(x_t^\star) \neq f_t(x_t^\pi)$ (otherwise we don't incur regret)

By the optimality of $x_t^\star$: $f_t(x_t^\star)'c^\star \leq f_t(x_t^\pi)'c^\star$

- That is, we can add constraint $(f_t(x_t^\star) - f_t(x_t^\pi))'c^\star \leq 0$ to $C(\mathcal{I}_{t+1})$

What Control Do We Have?

Let us assume $f_t(x_t^\star) \neq f_t(x_t^\pi)$ (otherwise we don't incur regret)

By the optimality of $x_t^\star$: $f_t(x_t^\star)' c^\star \leq f_t(x_t^\pi)' c^\star$

- That is, we can add constraint $(f_t(x_t^\star) - f_t(x_t^\pi))' c^\star \leq 0$ to $C(\mathcal{I}_{t+1})$

We do have some control over the vector $(f_t(x_t^\star) - f_t(x_t^\pi))$

What Control Do We Have?

Let us assume $f_t(x_t^\star) \neq f_t(x_t^\pi)$ (otherwise we don't incur regret)

By the optimality of $x_t^\star$: $f_t(x_t^\star)'c^\star \leq f_t(x_t^\pi)'c^\star$

- That is, we can add constraint $(f_t(x_t^\star) - f_t(x_t^\pi))'c^\star \leq 0$ to $C(\mathcal{I}_{t+1})$

We do have some control over the vector $(f_t(x_t^\star) - f_t(x_t^\pi))$

By the optimality of x_t^π: $f_t(x_t^\star)'c_t^\pi \geq f_t(x_t^\pi)'c_t^\pi$

- $(f_t(x_t^\star) - f_t(x_t^\pi))$ must satisfy $(f_t(x_t^\star) - f_t(x_t^\pi))'c_t^\pi \geq 0$

What Control Do We Have?

Let us assume $f_t(x_t^\star) \neq f_t(x_t^\pi)$ (otherwise we don't incur regret)

By the optimality of $x_t^\star$: $f_t(x_t^\star)'c^\star \leq f_t(x_t^\pi)'c^\star$

- That is, we can add constraint $(f_t(x_t^\star) - f_t(x_t^\pi))'c^\star \leq 0$ to $C(\mathcal{I}_{t+1})$

We do have some control over the vector $(f_t(x_t^\star) - f_t(x_t^\pi))$

By the optimality of x_t^π: $f_t(x_t^\star)'c_t^\pi \geq f_t(x_t^\pi)'c_t^\pi$

- $(f_t(x_t^\star) - f_t(x_t^\pi))$ must satisfy $(f_t(x_t^\star) - f_t(x_t^\pi))'c_t^\pi \geq 0$

The constraints $(f_t(x_t^\star) - f_t(x_t^\pi))'c^\star \leq 0$ and $(f_t(x_t^\star) - f_t(x_t^\pi))'c_t^\pi \geq 0$ jointly imply that either $c_t^\pi \notin C(\mathcal{I}_{t+1})$ or $c_t^\pi \in \partial C(\mathcal{I}_{t+1})$

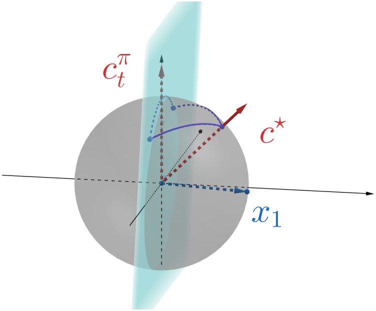
When We Don't Learn Much

For nature to cause regret in period t, it needs to either remove c_t^π from $C(\mathcal{I}_t)$ or at least cut the knowledge set through it

When We Don't Learn Much

For nature to cause regret in period t, it needs to either remove c_t^π from $C(\mathcal{I}_t)$ or at least cut the knowledge set through it

- Nature is able to cause significant and little learning when the proxy cost is at or near **the boundary** of C_0.



Inverse Exploration

We don't have any direct control over the information gain in our problem

Inverse Exploration

We don't have any direct control over the information gain in our problem

To cause regret, nature needs to cut c_t^π or move it to a boundary of $C(\mathcal{I}_{t+1})$

Inverse Exploration

We don't have any direct control over the information gain in our problem

To cause regret, nature needs to cut c_t^π or move it to a boundary of $C(\mathcal{I}_{t+1})$

If we choose c_t^π that is away from all the boundaries of $C(\mathcal{I}_t)$, nature needs to at least cut through c_t^π, giving us a lot of information

Inverse Exploration

We don't have any direct control over the information gain in our problem

To cause regret, nature needs to cut c_t^π or move it to a boundary of $C(\mathcal{I}_{t+1})$

If we choose c_t^π that is away from all the boundaries of $C(\mathcal{I}_t)$, nature needs to at least cut through c_t^π, giving us a lot of information

We call this process of forcing nature to choose between causing regret and impeding learning **inverse exploration**

The Circumcenter Trap

The circumcenter is the greedy policy (myopically optimal)

The Circumcenter Trap

The circumcenter is the greedy policy (myopically optimal)

The knowledge set evolves by incorporating new halfspace cuts

The Circumcenter Trap

The circumcenter is the greedy policy (myopically optimal)

The knowledge set evolves by incorporating new halfspace cuts

But the circumcenter of a polyhedral cone can easily lie on its boundary

The Circumcenter Trap

The circumcenter is the greedy policy (myopically optimal)

The knowledge set evolves by incorporating new halfspace cuts

But the circumcenter of a polyhedral cone can easily lie on its boundary

Once the circumcenter falls on the boundary, the circumcenter is trapped

The Circumcenter Trap

The circumcenter is the greedy policy (myopically optimal)

The knowledge set evolves by incorporating new halfspace cuts

But the circumcenter of a polyhedral cone can easily lie on its boundary

Once the circumcenter falls on the boundary, the circumcenter is trapped

We will solve this problem by regularizing the knowledge set.

The Circumcenter Trap

The circumcenter is the greedy policy (myopically optimal)

The knowledge set evolves by incorporating new halfspace cuts

But the circumcenter of a polyhedral cone can easily lie on its boundary

Once the circumcenter falls on the boundary, the circumcenter is trapped

We will solve this problem by regularizing the knowledge set.

- We will replace the knowledge sets by supersets that contain them

The Circumcenter Trap

The circumcenter is the greedy policy (myopically optimal)

The knowledge set evolves by incorporating new halfspace cuts

But the circumcenter of a polyhedral cone can easily lie on its boundary

Once the circumcenter falls on the boundary, the circumcenter is trapped

We will solve this problem by regularizing the knowledge set.

- We will replace the knowledge sets by supersets that contain them
- We will use **ellipsoidal cones**, which avoid this intertemporal tradeoff

The Circumcenter Trap

The circumcenter is the greedy policy (myopically optimal)

The knowledge set evolves by incorporating new halfspace cuts

But the circumcenter of a polyhedral cone can easily lie on its boundary

Once the circumcenter falls on the boundary, the circumcenter is trapped

We will solve this problem by regularizing the knowledge set.

- We will replace the knowledge sets by supersets that contain them
- We will use **ellipsoidal cones**, which avoid this intertemporal tradeoff
- Ellipsoidal cone: circumcenter = axis (farthest point from all borders)

The EllipsoidalCones Algorithm

EllipsoidalCones is a first step towards our final algorithm

The `EllipsoidalCones` Algorithm

`EllipsoidalCones` is a first step towards our final algorithm

- Choose c_t^{π} as the **circumcenter** of E_t

The `EllipsoidalCones` Algorithm

`EllipsoidalCones` is a first step towards our final algorithm

- Choose c_t^π as the **circumcenter** of E_t
- Choose action $x_t^\pi \in \arg\min_{x \in \mathcal{X}_t} f_t(x)' c_t^\pi$

The `EllipsoidalCones` Algorithm

`EllipsoidalCones` is a first step towards our final algorithm

- Choose c_t^π as the **circumcenter** of E_t
- Choose action $x_t^\pi \in \arg\min_{x \in \mathcal{X}_t} f_t(x)' c_t^\pi$
- Collect feedback $x_t^\star$ and compute $\delta_t^\pi = f_t(x_t^\pi) - f_t(x_t^\star)$

The `EllipsoidalCones` Algorithm

`EllipsoidalCones` is a first step towards our final algorithm

- Choose c_t^π as the **circumcenter** of E_t
- Choose action $x_t^\pi \in \arg\min_{x \in \mathcal{X}_t} f_t(x)' c_t^\pi$
- Collect feedback $x_t^\star$ and compute $\delta_t^\pi = f_t(x_t^\pi) - f_t(x_t^\star)$
- Update the ellipsoidal cone if δ_t^π is "informative in a new direction"

The `EllipsoidalCones` Algorithm

`EllipsoidalCones` is a first step towards our final algorithm

- Choose c_t^π as the **circumcenter** of E_t
- Choose action $x_t^\pi \in \arg\min_{x \in \mathcal{X}_t} f_t(x)' c_t^\pi$
- Collect feedback $x_t^\star$ and compute $\delta_t^\pi = f_t(x_t^\pi) - f_t(x_t^\star)$
- Update the ellipsoidal cone if δ_t^π is "informative in a new direction"

Types of periods:

- **No update**: we incur low regret

The `EllipsoidalCones` Algorithm

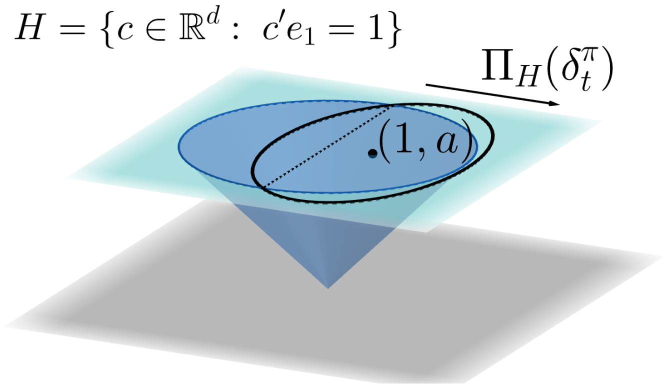
`EllipsoidalCones` is a first step towards our final algorithm

- Choose c_t^π as the **circumcenter** of E_t
- Choose action $x_t^\pi \in \arg\min_{x \in \mathcal{X}_t} f_t(x)' c_t^\pi$
- Collect feedback $x_t^\star$ and compute $\delta_t^\pi = f_t(x_t^\pi) - f_t(x_t^\star)$
- Update the ellipsoidal cone if δ_t^π is "informative in a new direction"

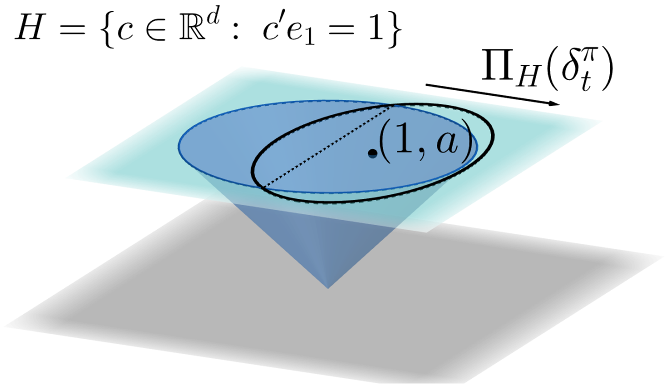
Types of periods:

- **No update**: we incur low regret
- **Cone update**: we gain valuable information about $c^\star$

Why Periods Without Updates?



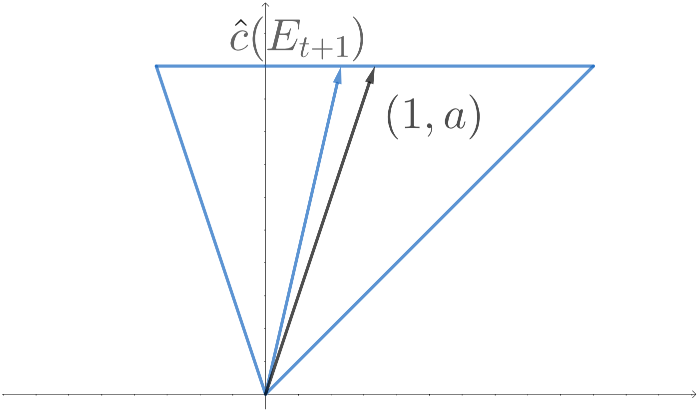
Why Periods Without Updates?



The ellipsoid method runs the risk of making the ellipsoid ill-conditioned (long and skinny). No-update periods prevent that from happening.

Ellipsoid Method for Ellipsoidal Cones

- The variation of the ellipsoid method we developed for cones is novel
- It required finding the best-fit new ellipsoidal cone after an update



Performance of `EllipsoidalCones`

Theorem

Consider any C_0 with $\alpha(C_0) < \pi/2$. Then, `EllipsoidalCones` *incurs regret:*

$$\mathrm{WCR}(C_0) = \mathcal{O}\left(d^2 \ln(T \tan \alpha(C_0))\right).$$

Performance of `EllipsoidalCones`

Theorem

Consider any C_0 with $\alpha(C_0) < \pi/2$. Then, `EllipsoidalCones` incurs regret:

$$\mathrm{WCR}(C_0) = \mathcal{O}\left(d^2 \ln(T \tan \alpha(C_0))\right).$$

First $\ln(T)$ regret bound for this problem.

Performance of `EllipsoidalCones`

Theorem

Consider any C_0 with $\alpha(C_0) < \pi/2$. Then, `EllipsoidalCones` incurs regret:

$$\mathrm{WCR}(C_0) = \mathcal{O}\left(d^2 \ln(T \tan \alpha(C_0))\right).$$

First $\ln(T)$ regret bound for this problem.

Requires that C_0 live inside a pointed cone.

Performance of EllipsoidalCones

Theorem

Consider any C_0 with $\alpha(C_0) < \pi/2$. Then, `EllipsoidalCones` *incurs regret:*

$$\mathrm{WCR}(C_0) = \mathcal{O}\left(d^2 \ln(T \tan \alpha(C_0))\right).$$

First $\ln(T)$ regret bound for this problem.

Requires that C_0 live inside a pointed cone.

- Can we relax this assumption?

What If we Started From a Nonpointed Set?

If $d = 1$ or 2, we reach a pointed set after 2 periods where $f_t(x_t^\pi) \neq f_t(x_t^\star)$

What If we Started From a Nonpointed Set?

If $d = 1$ or 2, we reach a pointed set after 2 periods where $f_t(x_t^\pi) \neq f_t(x_t^\star)$

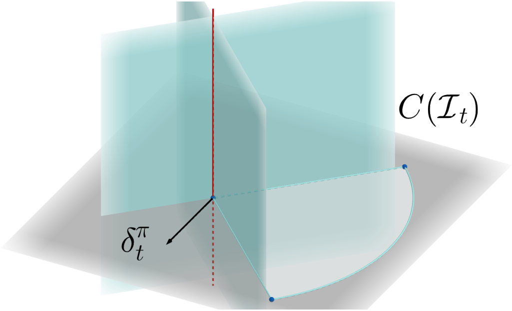
If $d \geq 3$, nature can stop the knowledge set from becoming pointed

What If we Started From a Nonpointed Set?

If $d = 1$ or 2, we reach a pointed set after 2 periods where $f_t(x_t^\pi) \neq f_t(x_t^\star)$

If $d \geq 3$, nature can stop the knowledge set from becoming pointed

This occurs if natures avoids 1 or more dimensions



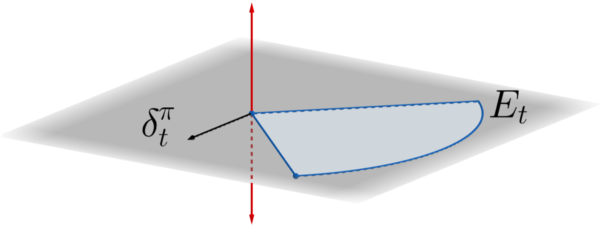
Ignoring Unused Dimensions

If nature decides not to use a dimension, we don't incur regret from it

Ignoring Unused Dimensions

If nature decides not to use a dimension, we don't incur regret from it

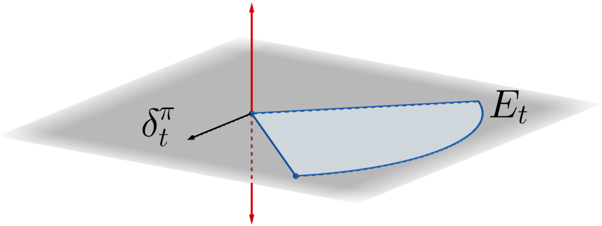
We can safely ignore such dimensions until nature decides to use them



Ignoring Unused Dimensions

If nature decides not to use a dimension, we don't incur regret from it

We can safely ignore such dimensions until nature decides to use them

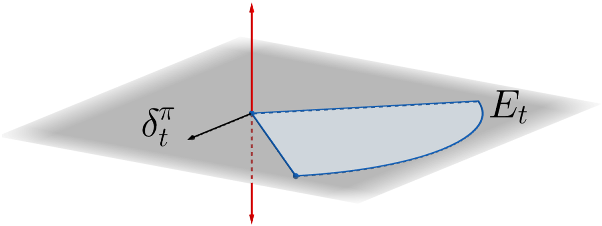


We do this by keeping track of subspace Δ_t where the projection of $C(\mathcal{I}_t)$ onto Δ_t lives inside a pointed cone

Ignoring Unused Dimensions

If nature decides not to use a dimension, we don't incur regret from it

We can safely ignore such dimensions until nature decides to use them



We do this by keeping track of subspace Δ_t where the projection of $C(\mathcal{I}_t)$ onto Δ_t lives inside a pointed cone

We ignore all information we have about costs orthogonal to Δ_t

The ProjectedCones Algorithm

- As long as we collect δ_t^π **close enough** to the subspace Δ_t, we proceed with a **robustified** version of the EllipsoidalCones

The `ProjectedCones` Algorithm

- As long as we collect δ_t^π **close enough** to the subspace Δ_t, we proceed with a **robustified** version of the `EllipsoidalCones`
- Otherwise we update Δ_{t+1} (increase the dimension) and fit a new cone

The `ProjectedCones` Algorithm

- As long as we collect δ_t^π **close enough** to the subspace Δ_t, we proceed with a **robustified** version of the `EllipsoidalCones`
- Otherwise we update Δ_{t+1} (increase the dimension) and fit a new cone

Types of periods:

The `ProjectedCones` Algorithm

- As long as we collect δ_t^π **close enough** to the subspace Δ_t, we proceed with a **robustified** version of the `EllipsoidalCones`
- Otherwise we update Δ_{t+1} (increase the dimension) and fit a new cone

Types of periods:

- **No update**: Low regret

The `ProjectedCones` Algorithm

- As long as we collect δ_t^π **close enough** to the subspace Δ_t, we proceed with a **robustified** version of the `EllipsoidalCones`
- Otherwise we update Δ_{t+1} (increase the dimension) and fit a new cone

Types of periods:

- **No update**: Low regret
- **Cone update**: Sufficient learning within the subspace

The `ProjectedCones` Algorithm

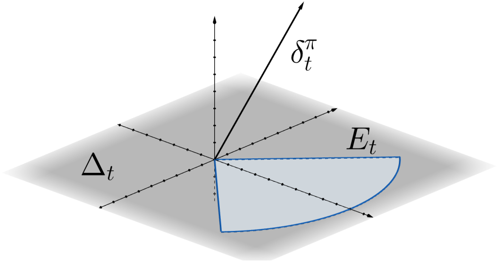
- As long as we collect δ_t^π **close enough** to the subspace Δ_t, we proceed with a **robustified** version of the `EllipsoidalCones`
- Otherwise we update Δ_{t+1} (increase the dimension) and fit a new cone

Types of periods:

- **No update**: Low regret
- **Cone update**: Sufficient learning within the subspace
- **Dimension update**: Construct a pointed cone in a higher dimension

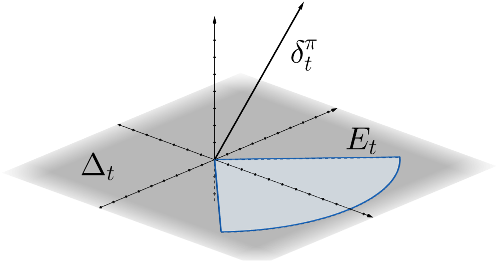
The Dimension Update

By performing a dimension update only if δ_t^π is sufficiently far from Δ_t, we obtain a higher-dimensional knowledge set that fits inside a pointed cone.



The Dimension Update

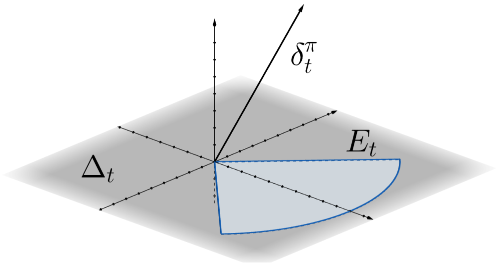
By performing a dimension update only if δ_t^π is sufficiently far from Δ_t, we obtain a higher-dimensional knowledge set that fits inside a pointed cone.



There is a tradeoff in how to set the minimum gap from Δ_t for an update.

The Dimension Update

By performing a dimension update only if δ_t^π is sufficiently far from Δ_t, we obtain a higher-dimensional knowledge set that fits inside a pointed cone.

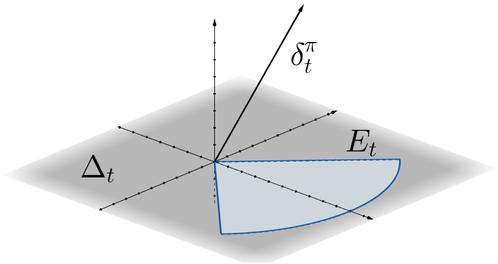


There is a tradeoff in how to set the minimum gap from Δ_t for an update.

- A bigger gap improves subspace updates (more pointed cone)

The Dimension Update

By performing a dimension update only if δ_t^π is sufficiently far from Δ_t, we obtain a higher-dimensional knowledge set that fits inside a pointed cone.



There is a tradeoff in how to set the minimum gap from Δ_t for an update.

- A bigger gap improves subspace updates (more pointed cone)
- A smaller gap improves cone updates (less robustness needed)

Performance of `ProjectedCones`

Theorem

For any C_0, `ProjectedCones` incurs regret:

$$\mathrm{WCR}(C_0) = \mathcal{O}\left(d^4 \ln T\right)$$

Summary of Online Learning Results

The circumcenter policy (optimal for offline) is a greedy policy.

Summary of Online Learning Results

The circumcenter policy (optimal for offline) is a greedy policy.

- We need a policy that forces nature to explore (inverse exploration).

Summary of Online Learning Results

The circumcenter policy (optimal for offline) is a greedy policy.

- We need a policy that forces nature to explore (inverse exploration).

We can make circumcenter work by making several adaptations:

Summary of Online Learning Results

The circumcenter policy (optimal for offline) is a greedy policy.

- We need a policy that forces nature to explore (inverse exploration).

We can make circumcenter work by making several adaptations:

- Replace polyhedral sets (bad for learning) with ellipsoidal cones.

Summary of Online Learning Results

The circumcenter policy (optimal for offline) is a greedy policy.

- We need a policy that forces nature to explore (inverse exploration).

We can make circumcenter work by making several adaptations:

- Replace polyhedral sets (bad for learning) with ellipsoidal cones.
- Adapt ellipsoid method to work with ellipsodial cones.

Summary of Online Learning Results

The circumcenter policy (optimal for offline) is a greedy policy.

- We need a policy that forces nature to explore (inverse exploration).

We can make circumcenter work by making several adaptations:

- Replace polyhedral sets (bad for learning) with ellipsoidal cones.
- Adapt ellipsoid method to work with ellipsodial cones.
- Skip knowledge set updates on low-regret periods.

Summary of Online Learning Results

The circumcenter policy (optimal for offline) is a greedy policy.

- We need a policy that forces nature to explore (inverse exploration).

We can make circumcenter work by making several adaptations:

- Replace polyhedral sets (bad for learning) with ellipsoidal cones.
- Adapt ellipsoid method to work with ellipsodial cones.
- Skip knowledge set updates on low-regret periods.
- Maintain subspace where knowledge set projection is pointed.

Summary of Online Learning Results

The circumcenter policy (optimal for offline) is a greedy policy.

- We need a policy that forces nature to explore (inverse exploration).

We can make circumcenter work by making several adaptations:

- Replace polyhedral sets (bad for learning) with ellipsoidal cones.
- Adapt ellipsoid method to work with ellipsodial cones.
- Skip knowledge set updates on low-regret periods.
- Maintain subspace where knowledge set projection is pointed.

First logarithmic regret bound for this class of models.

Takeaways

Feedback from optimal actions:

- Rich class of problems at the frontier of OR and ML

Takeaways

Feedback from optimal actions:

- Rich class of problems at the frontier of OR and ML
- This kind of feedback arises in a wide class of domains

Takeaways

Feedback from optimal actions:

- Rich class of problems at the frontier of OR and ML
- This kind of feedback arises in a wide class of domains
- Gives rise to a novel family of algorithms

Takeaways

Feedback from optimal actions:

- Rich class of problems at the frontier of OR and ML
- This kind of feedback arises in a wide class of domains
- Gives rise to a novel family of algorithms
- Imitation learning is quite different from statistical learning:

  inverse exploration vs. classical exploration-exploitation