CS-GY 6763: Lecture 9
Dimension Dependent Optimization: Center of Gravity & Ellipsoid

NYU, Prof. Ainesh Bakshi

First Order Convex Optimization

First Order Optimization: Given a convex function f and a convex set $\mathcal{S}$:

Goal: Find $\hat{\mathbf{x}} \in \mathcal{S}$ such that $f(\hat{\mathbf{x}}) \leq \min_{\mathbf{x} \in \mathcal{S}} f(\mathbf{x}) + \epsilon$.

Assume we have:

- **Function oracle**: Evaluate $f(\mathbf{x})$ for any $\mathbf{x}$.

First Order Convex Optimization

First Order Optimization: Given a convex function f and a convex set $\mathcal{S}$:

Goal: Find $\hat{\mathbf{x}} \in \mathcal{S}$ such that $f(\hat{\mathbf{x}}) \leq \min_{\mathbf{x} \in \mathcal{S}} f(\mathbf{x}) + \epsilon$.

Assume we have:

- **Function oracle**: Evaluate $f(\mathbf{x})$ for any $\mathbf{x}$.
- **Gradient oracle**: Evaluate $\nabla f(\mathbf{x})$ for any $\mathbf{x}$.

First Order Convex Optimization

First Order Optimization: Given a convex function f and a convex set $\mathcal{S}$:

Goal: Find $\hat{\mathbf{x}} \in \mathcal{S}$ such that $f(\hat{\mathbf{x}}) \leq \min_{\mathbf{x} \in \mathcal{S}} f(\mathbf{x}) + \epsilon$.

Assume we have:

- **Function oracle**: Evaluate $f(\mathbf{x})$ for any $\mathbf{x}$.
- **Gradient oracle**: Evaluate $\nabla f(\mathbf{x})$ for any $\mathbf{x}$.
- **Projection oracle**: Evaluate $P_{\mathcal{S}}(\mathbf{x})$ for any $\mathbf{x}$.

First Order Convex Optimization

First Order Optimization: Given a convex function f and a convex set $\mathcal{S}$:

Goal: Find $\hat{\mathbf{x}} \in \mathcal{S}$ such that $f(\hat{\mathbf{x}}) \leq \min_{\mathbf{x} \in \mathcal{S}} f(\mathbf{x}) + \epsilon$.

Assume we have:

- **Function oracle**: Evaluate $f(\mathbf{x})$ for any $\mathbf{x}$.
- **Gradient oracle**: Evaluate $\nabla f(\mathbf{x})$ for any $\mathbf{x}$.
- **Projection oracle**: Evaluate $P_{\mathcal{S}}(\mathbf{x})$ for any $\mathbf{x}$.

Gradient descent requires $O\left(\frac{R^2 G^2}{\epsilon^2}\right)$ oracle calls to solve the problem. We could only improve the ϵ dependence by making stronger assumptions on f (strong convexity, smoothness).

Dimension Dependent Bound

Let $f(\mathbf{x})$ be bounded between $[-B, B]$ on $\mathcal{S}$.

Theorem (Dimension Dependent Convex Optimization)

There is an algorithm (the Center-of-Gravity Method) which finds $\hat{\mathbf{x}}$ *satisfying* $f(\hat{\mathbf{x}}) \leq \min_{\mathbf{x}\in\mathcal{S}} f(\mathbf{x}) + \epsilon$ *using* $O(d\log(B/\epsilon))$ *calls to a function and gradient oracle for convex* f*.*

Dimension Dependent Bound

Let $f(\mathbf{x})$ be bounded between $[-B, B]$ on $\mathcal{S}$.

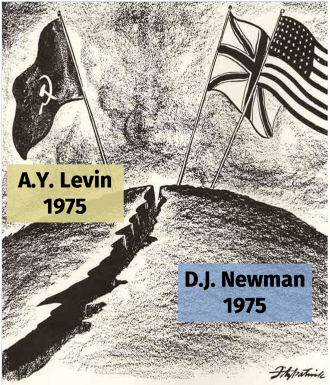
Theorem (Dimension Dependent Convex Optimization)

There is an algorithm (the Center-of-Gravity Method) which finds $\hat{\mathbf{x}}$ satisfying $f(\hat{\mathbf{x}}) \leq \min_{\mathbf{x}\in\mathcal{S}} f(\mathbf{x}) + \epsilon$ using $O(d \log(B/\epsilon))$ calls to a function and gradient oracle for convex f.

- **Caveat:** Assumes we have some representation of $\mathcal{S}$, not just a projection oracle.
- **Note:** For an unconstrained problem with known starting radius R, can take $\mathcal{S}$ to be the ball of radius R around $\mathbf{x}^{(0)}$. If $\|\nabla f(\mathbf{x})\|_2 \leq G$, we always have $B = O(RG)$.

Center of Gravity Method

Developed simultaneously on opposite sides of the Iron Curtain.



Not used in practice (oracle efficient but not computationally efficient) but the basic idea underlies many popular algorithms, including Ellipsoid Method.

Background

A few basic ingredients:

1. The center-of-gravity of a convex set $\mathcal{S}$ is defined as:

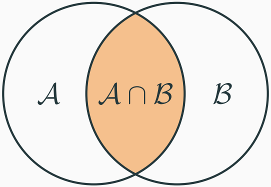
$$c = \frac{\int_{x \in \mathcal{S}} x \, dx}{\mathrm{vol}(\mathcal{S})} = \frac{\int_{x \in \mathcal{S}} x \, dx}{\int_{x \in \mathcal{S}} 1 \, dx}$$

A few basic ingredients:

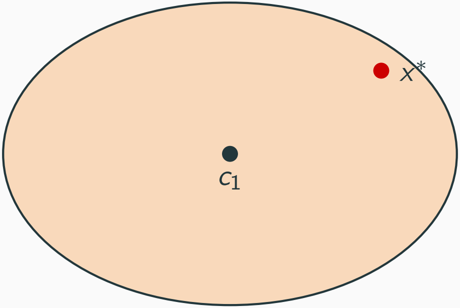
1. The center-of-gravity of a convex set $\mathcal{S}$ is defined as:

$$c = \frac{\int_{x \in \mathcal{S}} x \, dx}{\text{vol}(\mathcal{S})} = \frac{\int_{x \in \mathcal{S}} x \, dx}{\int_{x \in \mathcal{S}} 1 \, dx}$$

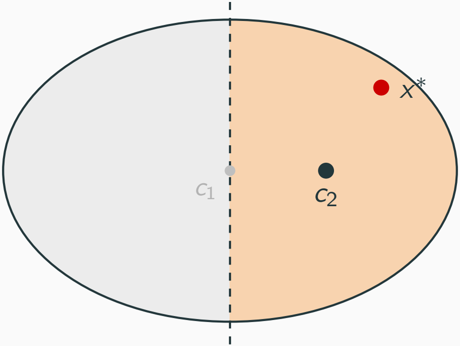
2. For two convex sets $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{A} \cap \mathcal{B}$ is convex. Proof by picture:



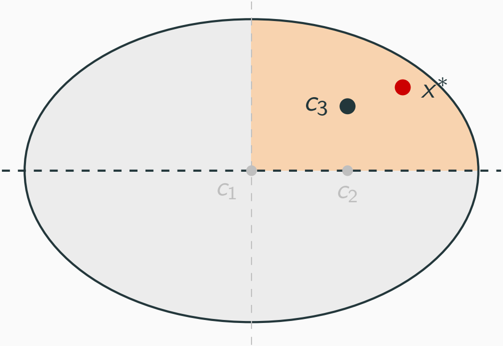
Cutting Plane Methods



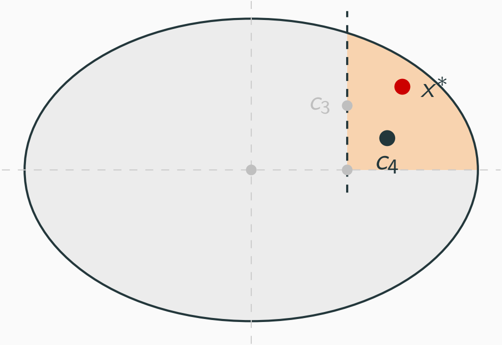
Center of Gravity Method



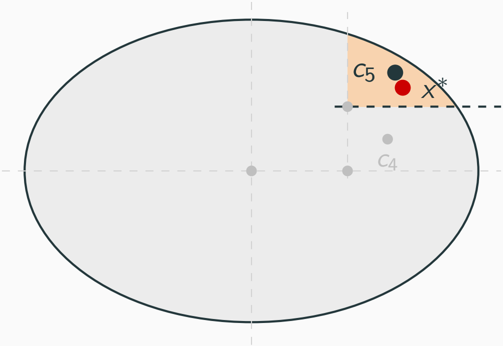
Center of Gravity Method



Center of Gravity Method



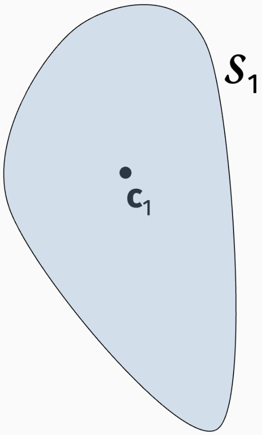
Center of Gravity Method



Center of Gravity Method

Natural "cutting plane" method.

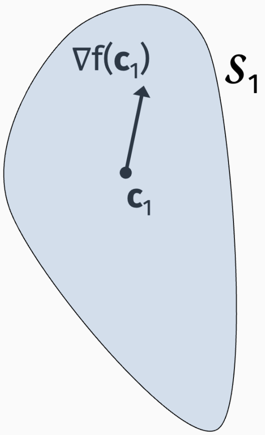
- $\mathcal{S}_1 = \mathcal{S}$
- For $t = 1, \ldots, T$:
  - $\mathbf{c}_t$ = center of gravity of $\mathcal{S}_t$.
  - Compute $\nabla f(\mathbf{c}_t)$.
  - $\mathcal{H} = \{\mathbf{x} \,|\, \langle \nabla f(\mathbf{c}_t), \mathbf{x} - \mathbf{c}_t \rangle \leq 0\}$.
  - $\mathcal{S}_{t+1} = \mathcal{S}_t \cap H$
- Return $\hat{\mathbf{x}} = \arg\min_{c_t} f(\mathbf{c}_t)$



Center of Gravity Method

Natural "cutting plane" method.

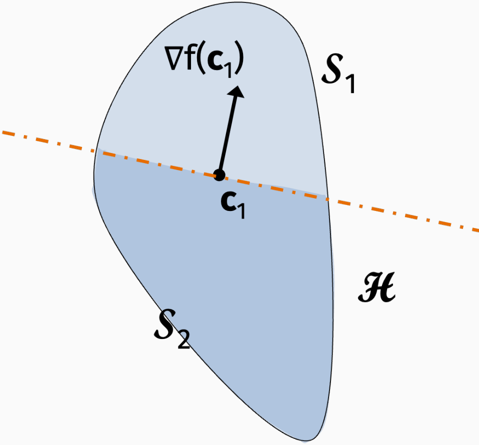
- $\mathcal{S}_1 = \mathcal{S}$
- For $t = 1, \ldots, T$:
  - $\mathbf{c}_t$ = center of gravity of $\mathcal{S}_t$.
  - Compute $\nabla f(\mathbf{c}_t)$.
  - $\mathcal{H} = \{\mathbf{x} \big| \langle \nabla f(\mathbf{c}_t), \mathbf{x} - \mathbf{c}_t \rangle \leq 0\}$.
  - $\mathcal{S}_{t+1} = \mathcal{S}_t \cap H$
- Return $\hat{\mathbf{x}} = \arg\min_{c_t} f(\mathbf{c}_t)$



Center of Gravity Method

Natural "cutting plane" method.

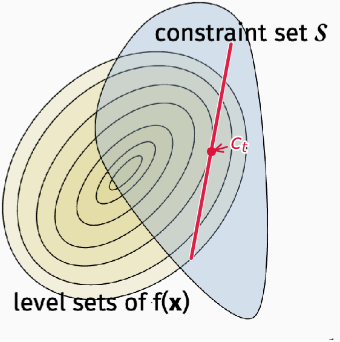
- $\mathcal{S}_1 = \mathcal{S}$
- For $t = 1, \ldots, T$:
  - $\mathbf{c}_t =$ center of gravity of $\mathcal{S}_t$.
  - Compute $\nabla f(\mathbf{c}_t)$.
  - $\mathcal{H} = \{\mathbf{x} \big| \langle \nabla f(\mathbf{c}_t), \mathbf{x} - \mathbf{c}_t \rangle \leq 0\}$.
  - $\mathcal{S}_{t+1} = \mathcal{S}_t \cap H$
- Return $\hat{\mathbf{x}} = \arg\min_{c_t} f(\mathbf{c}_t)$



Center of Gravity Method

Intuitively, why does it make sense to search in $\mathcal{S}_t \cap \mathcal{H}$ where:

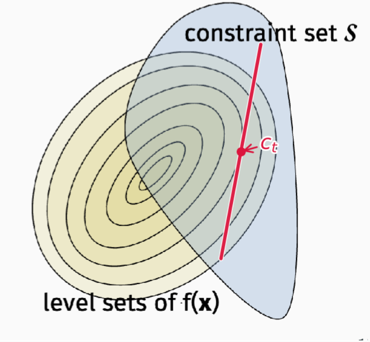
$$\mathcal{H} = \{\mathbf{x} \big| \langle \nabla f(\mathbf{c}_t), \mathbf{x} - \mathbf{c}_t \rangle \leq 0\}?$$



Center of Gravity Method

Intuitively, why does it make sense to search in $\mathcal{S}_t \cap \mathcal{H}$ where:

$$\mathcal{H} = \{\mathbf{x} \big| \langle \nabla f(\mathbf{c}_t), \mathbf{x} - \mathbf{c}_t \rangle \leq 0\}?$$



Only cuts off points that have higher function value than $f(\mathbf{c}_t)$, so we are guaranteed to never cut off the optimal point $\mathbf{x}^*$

Center of Gravity Method

Intuitively, why does it make sense to search in $\mathcal{S}_t \cap \mathcal{H}$ where:

$$\mathcal{H} = \{\mathbf{x} \big| \langle \nabla f(\mathbf{c}_t), \mathbf{x} - \mathbf{c}_t \rangle \leq 0\}?$$

By convexity,

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

Center of Gravity Method

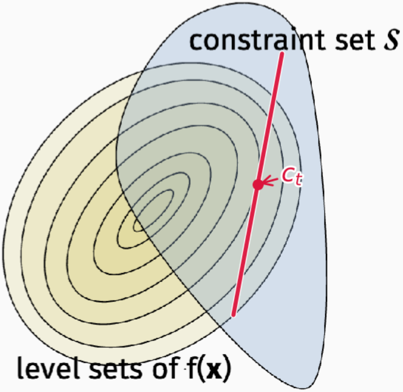
Intuitively, why does it make sense to search in $\mathcal{S}_t \cap \mathcal{H}$ where:

$$\mathcal{H} = \{\mathbf{x} \big| \langle \nabla f(\mathbf{c}_t), \mathbf{x} - \mathbf{c}_t \rangle \leq 0\}?$$

By convexity,

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

If $\mathbf{y} \notin \mathcal{H}$ then $\langle \nabla f(\mathbf{c}_t), \mathbf{y} - \mathbf{c}_t \rangle$ is positive, so $f(\mathbf{y}) > f(\mathbf{c}_t)$.



Convergence Theorem

Theorem (Center-of-Gravity Convergence)

Let f be a convex function with values in $[-B, B]$. Let $\hat{\mathbf{x}}$ be the output of the center-of-gravity method run for T iterations. Then:

$$f(\hat{\mathbf{x}}) - f(\mathbf{x}^*) \leq 2B\left(1 - \frac{1}{e}\right)^{T/d} \leq 2Be^{-T/3d}.$$

Convergence Theorem

Theorem (Center-of-Gravity Convergence)

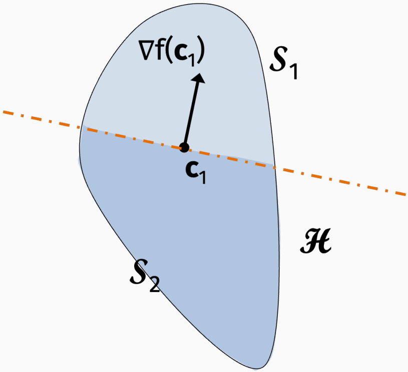
Let f be a convex function with values in $[-B, B]$. Let $\hat{\mathbf{x}}$ be the output of the center-of-gravity method run for T iterations. Then:

$$f(\hat{\mathbf{x}}) - f(\mathbf{x}^*) \leq 2B\left(1 - \frac{1}{e}\right)^{T/d} \leq 2Be^{-T/3d}.$$

If we set $T = 3d\log(2B/\epsilon)$, then $f(\hat{\mathbf{x}}) - f(\mathbf{x}^*) \leq \epsilon$.

Key Geometric Tool

Want to argue that, at every step of the algorithm, we "cut off" a large portion of the convex set we are searching over:



Key Geometric Tool

Theorem (Grünbaum's Theorem)

For any convex set $\mathcal{S}$ *with center-of-gravity* $\mathbf{c}$, *and any halfspace* $\mathcal{Z} = \{\mathbf{x} \big| \langle \mathbf{a}, \mathbf{x} - \mathbf{c} \rangle \leq 0\}$*:*

$$\frac{\text{vol}(\mathcal{S} \cap \mathcal{Z})}{\text{vol}(\mathcal{S})} \geq \frac{1}{e} \approx .368$$

Key Geometric Tool

Theorem (Grünbaum's Theorem)

For any convex set $\mathcal{S}$ *with center-of-gravity* $\mathbf{c}$*, and any halfspace* $\mathcal{Z} = \{\mathbf{x} \big| \langle \mathbf{a}, \mathbf{x} - \mathbf{c} \rangle \leq 0\}$*:*

$$\frac{\text{vol}(\mathcal{S} \cap \mathcal{Z})}{\text{vol}(\mathcal{S})} \geq \frac{1}{e} \approx .368$$

Let $\mathcal{Z}$ be the complement of $\mathcal{H}$ from the algorithm. Then we cut off at least a $1/e$ fraction of the convex body on every iteration.

Key Geometric Tool

Theorem (Grünbaum's Theorem)

For any convex set $\mathcal{S}$ *with center-of-gravity* $\mathbf{c}$*, and any halfspace* $\mathcal{Z} = \{\mathbf{x} \big| \langle \mathbf{a}, \mathbf{x} - \mathbf{c} \rangle \leq 0\}$*:*

$$\frac{\text{vol}(\mathcal{S} \cap \mathcal{Z})}{\text{vol}(\mathcal{S})} \geq \frac{1}{e} \approx .368$$

Let $\mathcal{Z}$ be the complement of $\mathcal{H}$ from the algorithm. Then we cut off at least a $1/e$ fraction of the convex body on every iteration.

Corollary: After t steps, $\text{vol}(\mathcal{S}_t) \leq \left(1 - \frac{1}{e}\right)^t \text{vol}(\mathcal{S})$.

Key Geometric Tool

Theorem (Grünbaum's Theorem)

For any convex set $\mathcal{S}$ *with center-of-gravity* $\mathbf{c}$*, and any halfspace* $\mathcal{Z} = \{\mathbf{x} \big| \langle \mathbf{a}, \mathbf{x} - \mathbf{c} \rangle \leq 0\}$*:*

$$\frac{\mathrm{vol}(\mathcal{S} \cap \mathcal{Z})}{\mathrm{vol}(\mathcal{S})} \geq \frac{1}{e} \approx .368$$

Let $\mathcal{Z}$ be the complement of $\mathcal{H}$ from the algorithm. Then we cut off at least a $1/e$ fraction of the convex body on every iteration.

Corollary: After t steps, $\mathrm{vol}(\mathcal{S}_t) \leq \left(1 - \frac{1}{e}\right)^t \mathrm{vol}(\mathcal{S})$.

Why?

Story So Far

- We have a cutting plane method which cuts off at least a $1/e$ fraction of the volume at each step.

Story So Far

- We have a cutting plane method which cuts off at least a $1/e$ fraction of the volume at each step.

- After T steps, $\text{vol}(\mathcal{S}_T) \leq \left(1 - \frac{1}{e}\right)^T \text{vol}(\mathcal{S})$.

Story So Far

- We have a cutting plane method which cuts off at least a $1/e$ fraction of the volume at each step.
- After T steps, $\text{vol}(\mathcal{S}_T) \leq \left(1 - \frac{1}{e}\right)^T \text{vol}(\mathcal{S})$.
- But we are interested in function values, not volumes.

Story So Far

- We have a cutting plane method which cuts off at least a $1/e$ fraction of the volume at each step.
- After T steps, $\mathrm{vol}(\mathcal{S}_T) \leq \left(1 - \frac{1}{e}\right)^T \mathrm{vol}(\mathcal{S})$.
- But we are interested in function values, not volumes.
- How do we connect the two?

Convergence Proof

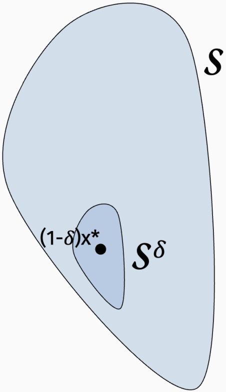
- Let δ be any small error parameter and let

  $\mathcal{S}^\delta = \{(1-\delta)\mathbf{x}^* + \delta\mathbf{x} \,\big|\, \text{for } \mathbf{x} \in \mathcal{S}\} = \{\mathbf{x}^* + \delta(\mathbf{x} - \mathbf{x}^*) \,\big|\, \text{for } \mathbf{x} \in \mathcal{S}\}$

Convergence Proof

- Let δ be any small error parameter and let

  $\mathcal{S}^\delta = \{(1-\delta)\mathbf{x}^* + \delta\mathbf{x} \mid \text{for } \mathbf{x} \in \mathcal{S}\} = \{\mathbf{x}^* + \delta(\mathbf{x} - \mathbf{x}^*) \mid \text{for } \mathbf{x} \in \mathcal{S}\}$
- This is a small "shrunk" version of $\mathcal{S}$ around the optimal point $\mathbf{x}^*$.

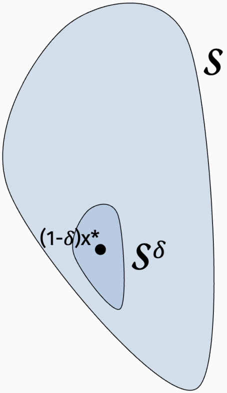


Convergence Proof

Claim 1: Every point $\mathbf{y}$ in $\mathcal{S}^{\delta}$ has good function value.

Convergence Proof

Claim 1: Every point **y** in $\mathcal{S}^\delta$ has good function value.

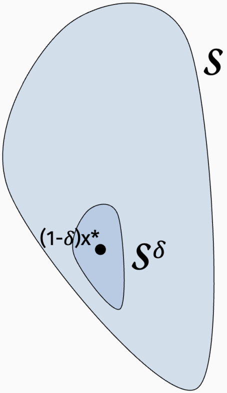


For any $\mathbf{y} \in \mathcal{S}^\delta$:

$$f(\mathbf{y}) = f\left((1-\delta)\mathbf{x}^* + \delta\mathbf{x}\right)$$

Convergence Proof

Claim 1: Every point $\mathbf{y}$ in $\mathcal{S}^\delta$ has good function value.

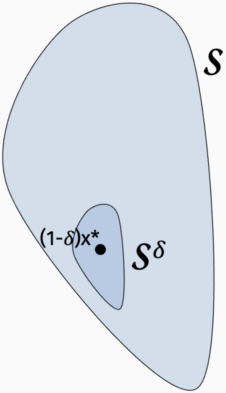


For any $\mathbf{y} \in \mathcal{S}^\delta$:

$$
\begin{aligned}
f(\mathbf{y}) &= f\left((1-\delta)\mathbf{x}^* + \delta\mathbf{x}\right) \\
&\leq (1-\delta)f(\mathbf{x}^*) + \delta f(\mathbf{x})
\end{aligned}
$$

Convergence Proof

Claim 1: Every point $\mathbf{y}$ in $\mathcal{S}^\delta$ has good function value.

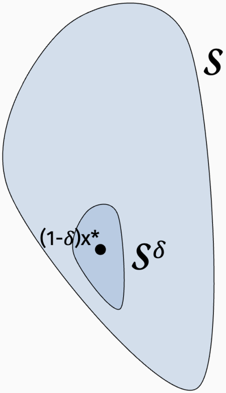


For any $\mathbf{y} \in \mathcal{S}^\delta$:

$$\begin{aligned} f(\mathbf{y}) &= f\left((1-\delta)\mathbf{x}^* + \delta\mathbf{x}\right) \\ &\leq (1-\delta)f(\mathbf{x}^*) + \delta f(\mathbf{x}) \\ &\leq f(\mathbf{x}^*) - \delta f(\mathbf{x}^*) + \delta f(\mathbf{x}) \end{aligned}$$

Convergence Proof

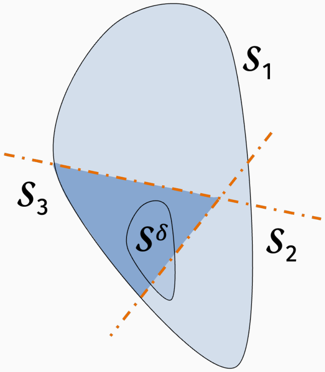
Claim 1: Every point $\mathbf{y}$ in $\mathcal{S}^\delta$ has good function value.



For any $\mathbf{y} \in \mathcal{S}^\delta$:

$$\begin{aligned} f(\mathbf{y}) &= f\left((1-\delta)\mathbf{x}^* + \delta\mathbf{x}\right) \\ &\leq (1-\delta)f(\mathbf{x}^*) + \delta f(\mathbf{x}) \\ &\leq f(\mathbf{x}^*) - \delta f(\mathbf{x}^*) + \delta f(\mathbf{x}) \\ &\leq f(\mathbf{x}^*) + 2B\delta. \end{aligned}$$

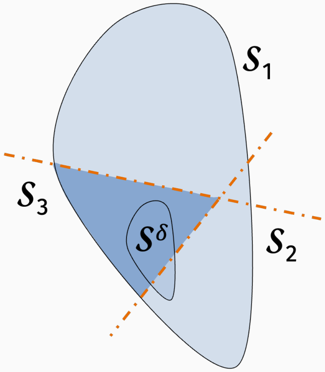
Convergence Proof



Claim 2: Let $\delta = (1 - \frac{1}{e})^{T/d}$. After T steps,

- Either we have that $\mathcal{S}_T$ equals S^δ exactly

Convergence Proof



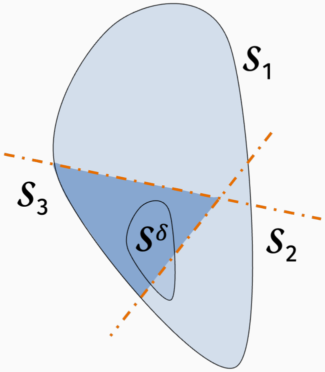
Claim 2: Let $\delta = (1 - \frac{1}{e})^{T/d}$. After T steps,

- Either we have that $\mathcal{S}_T$ equals S^δ exactly
- Or we "chopped off" at least one point $\mathbf{y}$ in $\mathcal{S}^\delta$.

Why?

- We never get rid of x^* as we chop off halfspaces, so $\mathcal{S}_t$ always contains x^*.

Convergence Proof



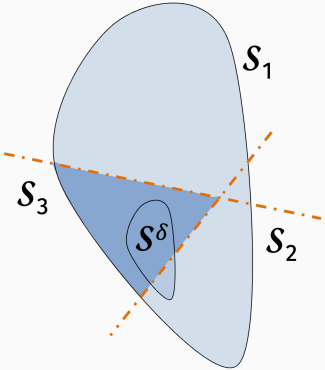
Claim 2: Let $\delta = (1 - \frac{1}{e})^{T/d}$. After T steps,

- Either we have that $\mathcal{S}_T$ equals S^δ exactly
- Or we "chopped off" at least one point $\mathbf{y}$ in $\mathcal{S}^\delta$.

Why?

- We never get rid of x^* as we chop off halfspaces, so $\mathcal{S}_t$ always contains x^*.
- If $\text{vol}(\mathcal{S}_t) \leq \text{vol}(\mathcal{S}^\delta)$, then the claim holds

Convergence Proof



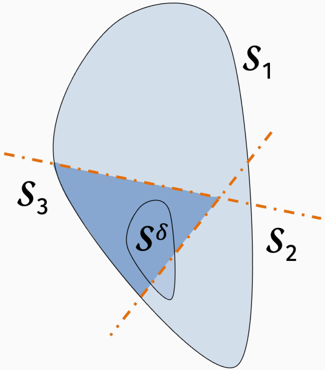
Claim 2: Let $\delta = (1 - \frac{1}{e})^{T/d}$. After T steps,

- Either we have that $\mathcal{S}_T$ equals S^δ exactly
- Or we "chopped off" at least one point $\mathbf{y}$ in $\mathcal{S}^\delta$.

Proof:

- $\mathcal{S}_t$ always contains x^*
- $\text{vol}(\mathcal{S}_t) \leq \left(1 - \frac{1}{e}\right)^T \text{vol}(\mathcal{S})$

Convergence Proof



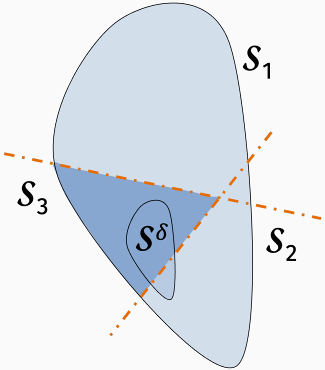
Claim 2: Let $\delta = (1 - \frac{1}{e})^{T/d}$. After T steps,

- Either we have that $\mathcal{S}_T$ equals S^δ exactly
- Or we "chopped off" at least one point $\mathbf{y}$ in $\mathcal{S}^\delta$.

Proof:

- $\mathcal{S}_t$ always contains x^*
- $\text{vol}(\mathcal{S}_t) \leq \left(1 - \frac{1}{e}\right)^T \text{vol}(\mathcal{S})$
- $\text{vol}(\mathcal{S}^\delta) = \delta^d \, \text{vol}(\mathcal{S})$

Convergence Proof



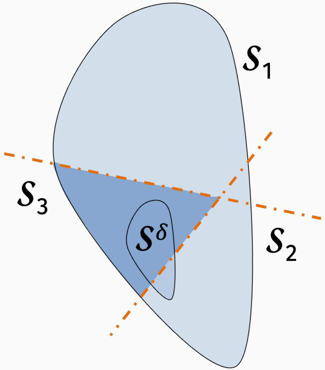
Claim 2: Let $\delta = (1 - \frac{1}{e})^{T/d}$. After T steps,

- Either we have that $\mathcal{S}_T$ equals S^δ exactly
- Or we "chopped off" at least one point $\mathbf{y}$ in $\mathcal{S}^\delta$.

Proof:

- $\mathcal{S}_t$ always contains x^*
- $\text{vol}(\mathcal{S}_t) \leq \left(1 - \frac{1}{e}\right)^T \text{vol}(\mathcal{S})$
- $\text{vol}(\mathcal{S}^\delta) = \delta^d \,\text{vol}(\mathcal{S})$
- For our choice of δ, $\text{vol}(\mathcal{S}_T) \leq \text{vol}(\mathcal{S}^\delta)$

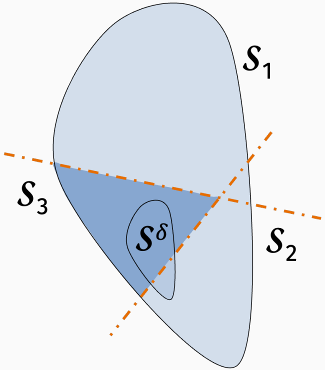
Convergence Proof



Claim 3: If $\mathcal{S}_T$ equals $\mathcal{S}^\delta$ or we "chopped off" at least one point $\mathbf{y}$ in $\mathcal{S}^\delta$, then:

$$f(\hat{\mathbf{x}}) = \min_{t=1,\ldots,T} f(\mathbf{c}_t) \leq f(\mathbf{x}^*) + 2B\delta.$$

Convergence Proof

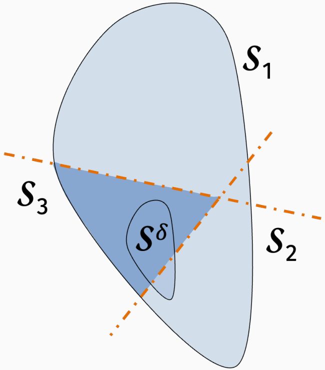


Claim 3: If $\mathcal{S}_T$ equals $\mathcal{S}^\delta$ or we "chopped off" at least one point $\mathbf{y}$ in $\mathcal{S}^\delta$, then:

$$f(\hat{\mathbf{x}}) = \min_{t=1,\ldots,T} f(\mathbf{c}_t) \leq f(\mathbf{x}^*) + 2B\delta.$$

Why?

Convergence Proof



Claim 3: If $\mathcal{S}_T$ equals S^δ or we "chopped off" at least one point $\mathbf{y}$ in $\mathcal{S}^\delta$, then:

$$f(\hat{\mathbf{x}}) = \min_{t=1,\ldots,T} f(\mathbf{c}_t) \leq f(\mathbf{x}^*) + 2B\delta.$$

Why?

- By Claim 1 all points in $\mathcal{S}^\delta$ have function value at most $f(\mathbf{x}^*) + 2B\delta$.
- If $\mathcal{S}_T$ equals S^δ, then $\hat{\mathbf{x}}$ is the center of gravity of $\mathcal{S}^\delta$
- If we "chopped off" at least one point $\mathbf{y}$ in $\mathcal{S}^\delta$, then for some centroid $\mathbf{c}_t$, $f(\mathbf{c}_t) \leq f(\mathbf{y}) \leq f(\mathbf{x}^*) + 2B\delta$.

Convergence Theorem

Theorem (Center-of-Gravity Convergence)

Let f be a convex function with values in $[-B, B]$. Let $\hat{\mathbf{x}}$ be the output of the center-of-gravity method run for T iterations. Then:

$$f(\hat{\mathbf{x}}) - f(\mathbf{x}^*) \leq 2B\left(1 - \frac{1}{e}\right)^{T/d} \leq 2Be^{-T/3d}.$$

Convergence Theorem

Theorem (Center-of-Gravity Convergence)

Let f be a convex function with values in $[-B, B]$. Let $\hat{\mathbf{x}}$ be the output of the center-of-gravity method run for T iterations. Then:

$$f(\hat{\mathbf{x}}) - f(\mathbf{x}^*) \leq 2B\left(1 - \frac{1}{e}\right)^{T/d} \leq 2Be^{-T/3d}.$$

If we set $T = O\left(d \log(B/\epsilon)\right)$, then $f(\hat{\mathbf{x}}) - f(\mathbf{x}^*) \leq \epsilon$.

Convergence Theorem

Theorem (Center-of-Gravity Convergence)

Let f be a convex function with values in $[-B, B]$. Let $\hat{\mathbf{x}}$ be the output of the center-of-gravity method run for T iterations. Then:

$$f(\hat{\mathbf{x}}) - f(\mathbf{x}^*) \leq 2B\left(1 - \frac{1}{e}\right)^{T/d} \leq 2Be^{-T/3d}.$$

If we set $T = O(d \log(B/\epsilon))$, then $f(\hat{\mathbf{x}}) - f(\mathbf{x}^*) \leq \epsilon$.

In terms of gradient-oracle complexity, this is essentially optimal.
So why isn't the algorithm used?

Runtime Issue

- **Computing the centroid is hard in general** — #P-hard even when $\mathcal{S}$ is an intersection of half-spaces (a polytope).

Runtime Issue

- **Computing the centroid is hard in general** — #P-hard even when $\mathcal{S}$ is an intersection of half-spaces (a polytope).
- Even if the problem isn't hard for the starting convex body $\mathcal{S}$, it likely will become hard for $\mathcal{S} \cap \mathcal{H}_1 \cap \mathcal{H}_2 \cap \mathcal{H}_3 \cdots$.

Runtime Issue

- **Computing the centroid is hard in general** — #P-hard even when $\mathcal{S}$ is an intersection of half-spaces (a polytope).
- Even if the problem isn't hard for the starting convex body $\mathcal{S}$, it likely will become hard for $\mathcal{S} \cap \mathcal{H}_1 \cap \mathcal{H}_2 \cap \mathcal{H}_3 \cdots$.
- So while the oracle complexity of dimension-dependent optimization was settled in the 1960s, basic questions remained regarding computational complexity.

Runtime Issue

- **Computing the centroid is hard in general** — #P-hard even when $\mathcal{S}$ is an intersection of half-spaces (a polytope).
- Even if the problem isn't hard for the starting convex body $\mathcal{S}$, it likely will become hard for $\mathcal{S} \cap \mathcal{H}_1 \cap \mathcal{H}_2 \cap \mathcal{H}_3 \cdots$.
- So while the oracle complexity of dimension-dependent optimization was settled in the 1960s, basic questions remained regarding computational complexity.
- We will resolve this with an elegant cutting-plane method called the **Ellipsoid Method**, introduced by Naum Shor in 1977.

Formalization

To talk about runtime efficiency we need to be more concrete about how our (convex) constraint set is specified.

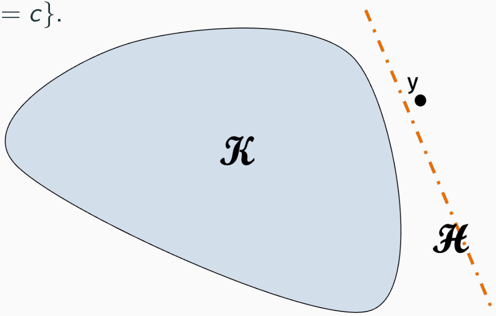
Formalization

To talk about runtime efficiency we need to be more concrete about how our (convex) constraint set is specified.

Separation Oracle: For a convex set $\mathcal{K} \subset \mathbb{R}^d$, a separation oracle $\mathbf{S}_{\mathcal{K}}$ is a function that takes in points in $\mathbb{R}^d$ and returns:

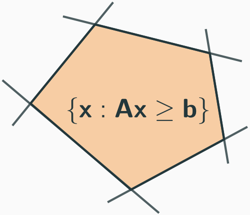
$$\mathbf{S}_{\mathcal{K}}(\mathbf{y}) = \begin{cases} \emptyset & \text{if } \mathbf{y} \in \mathcal{K}. \\ \text{separating hyperplane } (\mathbf{a}, c) & \text{if } \mathbf{y} \notin \mathcal{K}. \end{cases}$$

$\mathcal{H} = \{\mathbf{x} : \mathbf{a}^T \mathbf{x} = c\}$.



Separation Oracle

Example: How would you implement a separation oracle for a polytope $\{\mathbf{x} : \mathbf{A}\mathbf{x} \geq \mathbf{b}\}$?

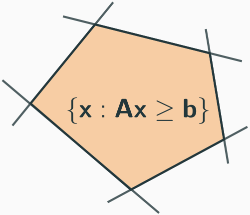


Each constraint $\mathbf{a}_i^T \mathbf{x} \geq b_i$ defines one halfspace. Given a query point $\mathbf{y}$:

- If all constraints satisfied: $\mathbf{y} \in \mathcal{K}$.

Separation Oracle

Example: How would you implement a separation oracle for a polytope $\{\mathbf{x} : \mathbf{A}\mathbf{x} \geq \mathbf{b}\}$?

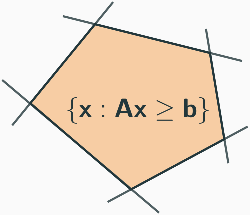


Each constraint $\mathbf{a}_i^T \mathbf{x} \geq b_i$ defines one halfspace. Given a query point $\mathbf{y}$:

- If all constraints satisfied: $\mathbf{y} \in \mathcal{K}$.
- Else there is a violated constraint such that $\mathbf{a}_i^T \mathbf{y} < b_i$.

Separation Oracle

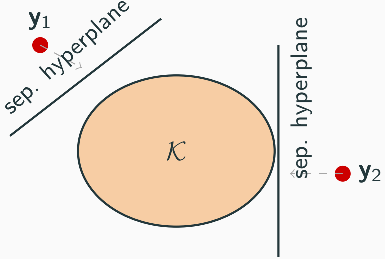
Example: How would you implement a separation oracle for a polytope $\{\mathbf{x} : \mathbf{A}\mathbf{x} \geq \mathbf{b}\}$?



Each constraint $\mathbf{a}_i^T\mathbf{x} \geq b_i$ defines one halfspace. Given a query point $\mathbf{y}$:

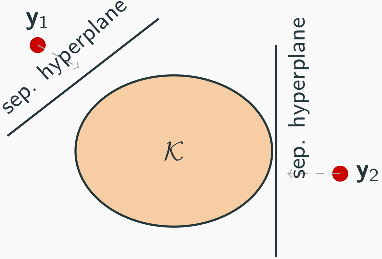
- If all constraints satisfied: $\mathbf{y} \in \mathcal{K}$.
- Else there is a violated constraint such that $\mathbf{a}_i^T\mathbf{y} < b_i$.
- But $\mathbf{a}_i^T\mathbf{x} \geq b_i$ for all $\mathbf{x} \in \mathcal{K}$, so $\mathbf{a}_i^T\mathbf{x} = b_i$ is a separating hyperplane.

Problem Simplification



Instead of solving a constrained optimization problem directly, solve the membership problem.

Problem Simplification



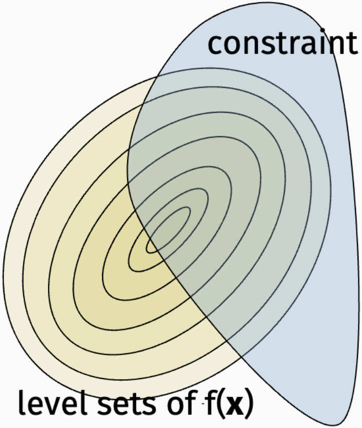
Instead of solving a constrained optimization problem directly, solve the membership problem.

Membership Problem: Given a separation oracle $S_{\mathcal{K}}$ for a convex set $\mathcal{K}$, determine if $\mathcal{K}$ is empty, or return any $\mathbf{x} \in \mathcal{K}$.

Recall, for each $\mathbf{y} \notin \mathcal{K}$, the oracle returns a separating hyperplane certifying that $\mathbf{y}$ is outside.

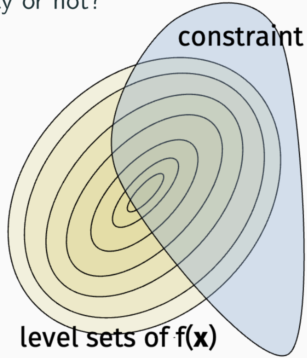
Original problem: $\min_{\mathbf{x} \in \mathcal{S}} f(\mathbf{x})$.

How to reduce to determining if a convex set $\mathcal{K}$ is empty or not?



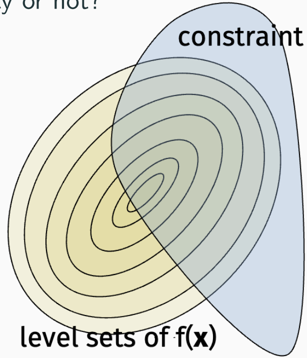
From Membership to Optimization

Original problem: $\min_{\mathbf{x} \in \mathcal{S}} f(\mathbf{x})$. How to reduce to determining if a convex set $\mathcal{K}$ is empty or not?



From Membership to Optimization

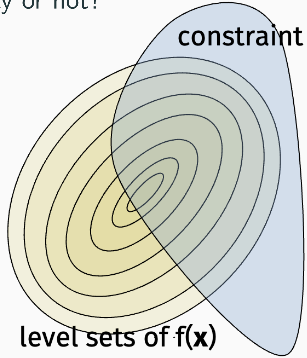
Original problem: $\min_{\mathbf{x}\in\mathcal{S}} f(\mathbf{x})$. How to reduce to determining if a convex set $\mathcal{K}$ is empty or not?



Claim: Given any fixed value c, can check if $f(\mathbf{x}^*) \leq c$ and, if so, find some $\mathbf{x}$ with $f(\mathbf{x}) \leq c$.

From Membership to Optimization

Original problem: $\min_{\mathbf{x}\in\mathcal{S}} f(\mathbf{x})$. How to reduce to determining if a convex set $\mathcal{K}$ is empty or not?

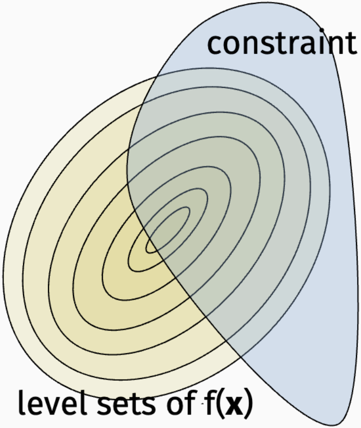


Claim: Given any fixed value c, can check if $f(\mathbf{x}^*) \leq c$ and, if so, find some $\mathbf{x}$ with $f(\mathbf{x}) \leq c$.

Approach: Solve membership problem on $\mathcal{K} = \mathcal{S} \cap \mathcal{C}$ where $\mathcal{C} = \{\mathbf{x} : f(\mathbf{x}) \leq c\}$. $\mathcal{C}$ and $\mathcal{S}$ are convex, so $\mathcal{K}$ is as well.

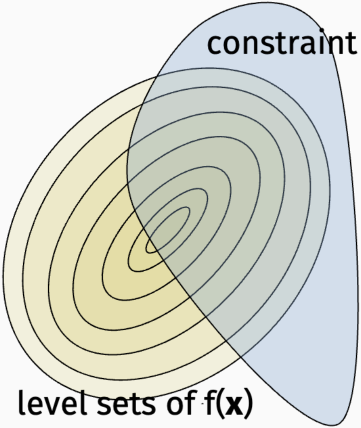
From Membership to Optimization

Prove on homework: Given efficient separation oracles for $\mathcal{C}$ and $\mathcal{S}$, I can construct an efficient separation oracle for $\mathcal{K}$.



From Membership to Optimization

Prove on homework: Given efficient separation oracles for $\mathcal{C}$ and $\mathcal{S}$, I can construct an efficient separation oracle for $\mathcal{K}$.

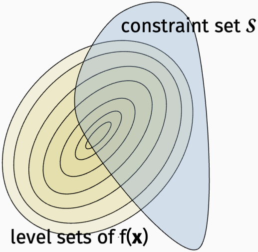


How do I get a separation oracle for $\mathcal{C}$?

From Membership to Optimization

How do I get a separation oracle for $\mathcal{C}$?

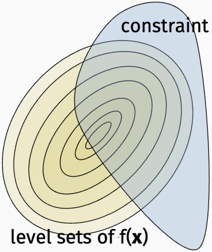
- Check if $f(\mathbf{y}) \leq c$.



From Membership to Optimization

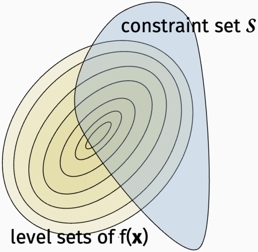
How do I get a separation oracle for $\mathcal{C}$?

- Check if $f(\mathbf{y}) \leq c$.
- If $\mathbf{y} \in \mathcal{C}$ and we are done



From Membership to Optimization

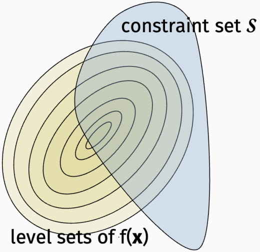
How do I get a separation oracle for $\mathcal{C}$?



- Check if $f(\mathbf{y}) \leq c$.
- If $\mathbf{y} \in \mathcal{C}$ and we are done
- Otherwise compute $\nabla f(\mathbf{y})$ and return the hyperplane $\mathcal{H} = \{\mathbf{x} : \nabla f(\mathbf{y})^T(\mathbf{x} - \mathbf{y}) = 0\}$

From Membership to Optimization

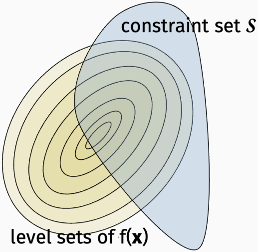
How do I get a separation oracle for $\mathcal{C}$?



- Check if $f(\mathbf{y}) \leq c$.
- If $\mathbf{y} \in \mathcal{C}$ and we are done
- Otherwise compute $\nabla f(\mathbf{y})$ and return the hyperplane $\mathcal{H} = \{\mathbf{x} : \nabla f(\mathbf{y})^T(\mathbf{x} - \mathbf{y}) = 0\}$
- **Why does this work?**
  - For all $\mathbf{w} \in \mathcal{C}$, $f(\mathbf{w}) \leq c < f(y)$.

From Membership to Optimization

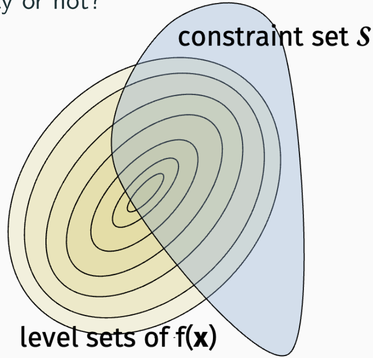
How do I get a separation oracle for $\mathcal{C}$?



- Check if $f(\mathbf{y}) \leq c$.
- If $\mathbf{y} \in \mathcal{C}$ and we are done
- Otherwise compute $\nabla f(\mathbf{y})$ and return the hyperplane $\mathcal{H} = \{\mathbf{x} : \nabla f(\mathbf{y})^T(\mathbf{x} - \mathbf{y}) = 0\}$
- **Why does this work?**
  - For all $\mathbf{w} \in \mathcal{C}$, $f(\mathbf{w}) \leq c < f(y)$.
  - If $\mathbf{z} : \nabla f(\mathbf{y})^T(\mathbf{z} - \mathbf{y}) \geq 0$, then by convexity $f(\mathbf{z}) \geq f(\mathbf{y}) > f(\mathbf{w})$ so $\mathbf{z} \notin \mathcal{C}$.

From Membership to Optimization

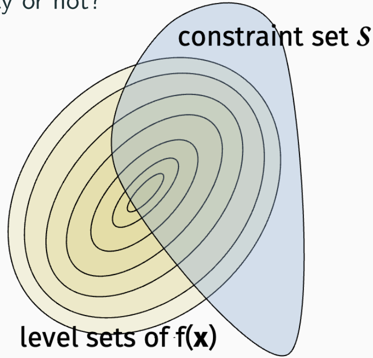
Original problem: $\min_{\mathbf{x}\in\mathcal{S}} f(\mathbf{x})$. How to reduce to determining if a convex set $\mathcal{K}$ is empty or not?



Claim: Given any fixed value c, can check if $f(\mathbf{x}^*) \leq c$ and, if so, find some $\mathbf{x}$ with $f(\mathbf{x}) \leq c$.

From Membership to Optimization

Original problem: $\min_{\mathbf{x}\in\mathcal{S}} f(\mathbf{x})$. How to reduce to determining if a convex set $\mathcal{K}$ is empty or not?



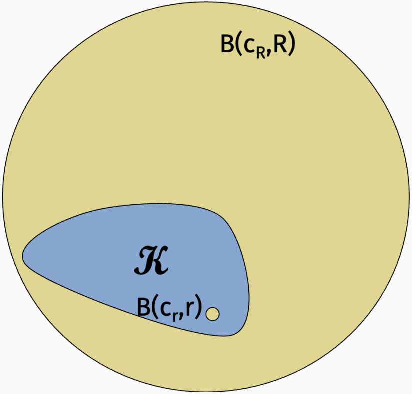
Claim: Given any fixed value c, can check if $f(\mathbf{x}^*) \leq c$ and, if so, find some $\mathbf{x}$ with $f(\mathbf{x}) \leq c$.

Final algorithm: Assuming f is positive, just run exponential/binary search to find $\tilde{c} \leq f(\mathbf{x}^*) + \epsilon$!

Ellipsoid Method Sketch

Goal of ellipsoid algorithm: Solve "Is $\mathcal{K}$ empty or not?" given a separation oracle for $\mathcal{K}$ under the assumptions that:

1. $\mathcal{K} \subset B(\mathbf{c}_R, R)$.
2. If non-empty, $\mathcal{K}$ contains $B(\mathbf{c}_r, r)$ for some $r < R$.



Ellipsoid Method Sketch

Iterative method similar to center-of-gravity:

1. Check if center $\mathbf{c}_R$ of $B(\mathbf{c}_R, R)$ is in $\mathcal{K}$.

Ellipsoid Method Sketch

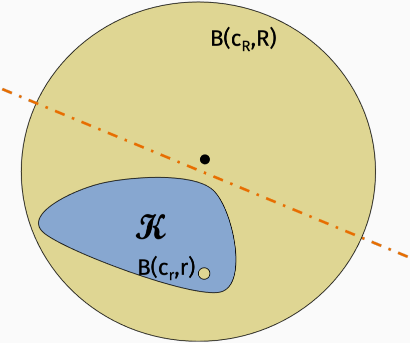
Iterative method similar to center-of-gravity:

1. Check if center $\mathbf{c}_R$ of $B(\mathbf{c}_R, R)$ is in $\mathcal{K}$.
2. If it is, we are done.

Ellipsoid Method Sketch

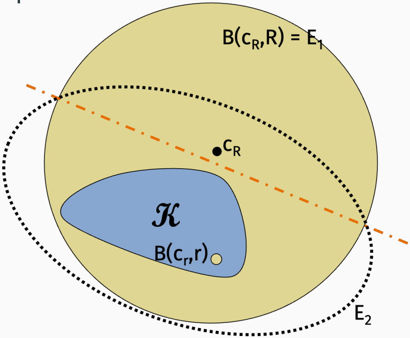
Iterative method similar to center-of-gravity:

1. Check if center $\mathbf{c}_R$ of $B(\mathbf{c}_R, R)$ is in $\mathcal{K}$.
2. If it is, we are done.
3. If not, cut search space in half using a separating hyperplane.



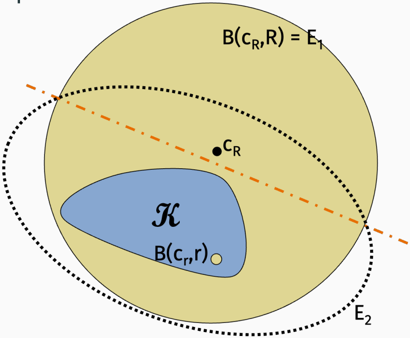
Ellipsoid Method Sketch

Key insight: Before moving on, approximate the new search region by something whose centroid is easy to compute — specifically, an ellipse!



Ellipsoid Method Sketch

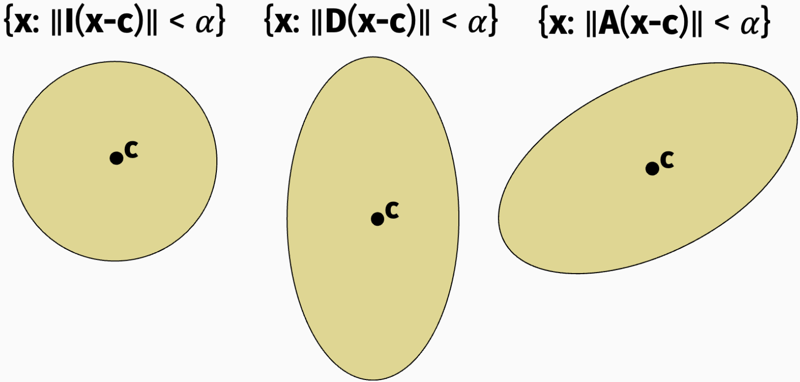
Key insight: Before moving on, approximate the new search region by something whose centroid is easy to compute — specifically, an ellipse!



Produce a sequence of ellipses that always contain $\mathcal{K}$ and decrease in volume: $B(\mathbf{c}_R, R) = E_1, E_2, \ldots$. Once we get to an ellipse with volume $\leq B(\mathbf{c}_r, r)$, we know $\mathcal{K}$ must be empty.

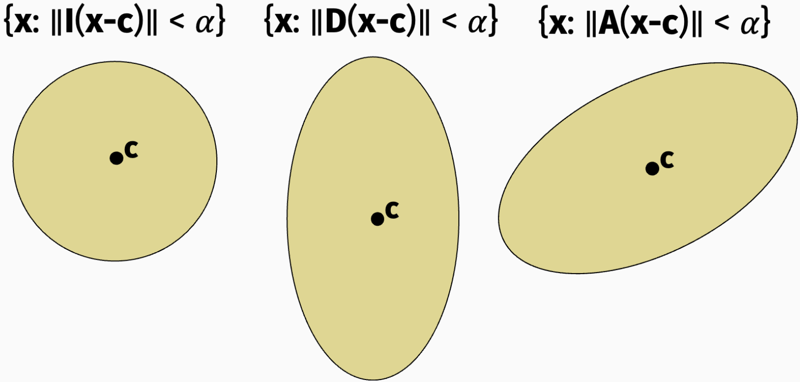
Ellipse

An ellipse is a convex set of the form $\{\mathbf{x} : \|\mathbf{A}(\mathbf{x} - \mathbf{c})\|_2^2 \leq \alpha\}$ for some constant α and matrix $\mathbf{A}$. The center-of-mass is $\mathbf{c}$.



Ellipse

An ellipse is a convex set of the form $\{\mathbf{x} : \|\mathbf{A}(\mathbf{x} - \mathbf{c})\|_2^2 \leq \alpha\}$ for some constant α and matrix $\mathbf{A}$. The center-of-mass is $\mathbf{c}$.



Often re-parameterized as $\{\mathbf{x} : (\mathbf{x} - \mathbf{c})^T \mathbf{Q}^{-1} (\mathbf{x} - \mathbf{c}) \leq 1\}$.

Ellipse

An ellipse is a convex set of the form $\{\mathbf{x} : \|\mathbf{A}(\mathbf{x} - \mathbf{c})\|_2^2 \leq \alpha\}$ for some constant α and matrix $\mathbf{A}$. The center-of-mass is $\mathbf{c}$.

Often re-parameterized as $\{\mathbf{x} : (\mathbf{x} - \mathbf{c})^T \mathbf{Q}^{-1} (\mathbf{x} - \mathbf{c}) \leq 1\}$.

Why are these two equivalent?

$$\|\mathbf{A}(\mathbf{x} - \mathbf{c})\|_2^2 = \langle \mathbf{A}(\mathbf{x} - \mathbf{c}), \mathbf{A}(\mathbf{x} - \mathbf{c}) \rangle$$

Ellipse

An ellipse is a convex set of the form $\{\mathbf{x} : \|\mathbf{A}(\mathbf{x}-\mathbf{c})\|_2^2 \leq \alpha\}$ for some constant α and matrix $\mathbf{A}$. The center-of-mass is $\mathbf{c}$.

Often re-parameterized as $\{\mathbf{x} : (\mathbf{x}-\mathbf{c})^T\mathbf{Q}^{-1}(\mathbf{x}-\mathbf{c}) \leq 1\}$.

Why are these two equivalent?

$$\begin{aligned}\|\mathbf{A}(\mathbf{x}-\mathbf{c})\|_2^2 &= \langle \mathbf{A}(\mathbf{x}-\mathbf{c}), \mathbf{A}(\mathbf{x}-\mathbf{c})\rangle \\ &= (\mathbf{x}-\mathbf{c})^T\mathbf{A}^T\mathbf{A}(\mathbf{x}-\mathbf{c})\end{aligned}$$

Ellipse

An ellipse is a convex set of the form $\{\mathbf{x} : \|\mathbf{A}(\mathbf{x}-\mathbf{c})\|_2^2 \leq \alpha\}$ for some constant α and matrix $\mathbf{A}$. The center-of-mass is $\mathbf{c}$.

Often re-parameterized as $\{\mathbf{x} : (\mathbf{x}-\mathbf{c})^T\mathbf{Q}^{-1}(\mathbf{x}-\mathbf{c}) \leq 1\}$.

Why are these two equivalent?

$$\begin{aligned}\|\mathbf{A}(\mathbf{x}-\mathbf{c})\|_2^2 &= \langle \mathbf{A}(\mathbf{x}-\mathbf{c}), \mathbf{A}(\mathbf{x}-\mathbf{c})\rangle \\ &= (\mathbf{x}-\mathbf{c})^T\mathbf{A}^T\mathbf{A}(\mathbf{x}-\mathbf{c})\end{aligned}$$

Ellipse

An ellipse is a convex set of the form $\{\mathbf{x} : \|\mathbf{A}(\mathbf{x}-\mathbf{c})\|_2^2 \leq \alpha\}$ for some constant α and matrix $\mathbf{A}$. The center-of-mass is $\mathbf{c}$.

Often re-parameterized as $\{\mathbf{x} : (\mathbf{x}-\mathbf{c})^T\mathbf{Q}^{-1}(\mathbf{x}-\mathbf{c}) \leq 1\}$.

Why are these two equivalent?

$$\begin{aligned}\|\mathbf{A}(\mathbf{x}-\mathbf{c})\|_2^2 &= \langle \mathbf{A}(\mathbf{x}-\mathbf{c}), \mathbf{A}(\mathbf{x}-\mathbf{c})\rangle \\ &= (\mathbf{x}-\mathbf{c})^T\mathbf{A}^T\mathbf{A}(\mathbf{x}-\mathbf{c})\end{aligned}$$

Set $\mathbf{Q} = \frac{1}{\alpha}(\mathbf{A}^T\mathbf{A})^{-1}$

Ellipsoid Update

Let $\mathbf{E}_i$ have parameters $\mathbf{Q}_i, \mathbf{c}_i$ and consider the half-ellipse obtained by cutting $\mathbf{E}_i$ with the hyperplane through the center $\mathbf{c}_i$:

$$\mathbf{E}_i \cap \{\mathbf{x} : \mathbf{a}_i^T \mathbf{x} \leq \mathbf{a}_i^T \mathbf{c}_i\}.$$

Key Insight: There is a closed form solution for the smallest ellipse containing a given half-ellipse!

Ellipsoid Update

Let $\mathbf{E}_i$ have parameters $\mathbf{Q}_i, \mathbf{c}_i$ and consider the half-ellipse obtained by cutting $\mathbf{E}_i$ with the hyperplane through the center $\mathbf{c}_i$:

$$\mathbf{E}_i \cap \{\mathbf{x} : \mathbf{a}_i^T \mathbf{x} \leq \mathbf{a}_i^T \mathbf{c}_i\}.$$

Key Insight: There is a closed form solution for the smallest ellipse containing a given half-ellipse!

Then $\mathbf{E}_{i+1}$ is the ellipse with parameters:

$$\mathbf{Q}_{i+1} = \frac{d^2}{d^2 - 1}\left(\mathbf{Q}_i - \frac{2}{d+1}\mathbf{h}\mathbf{h}^T\right) \qquad \mathbf{c}_{i+1} = \mathbf{c}_i - \frac{1}{d+1}\mathbf{h},$$

where $\mathbf{h} = \sqrt{\mathbf{a}_i^T \mathbf{Q}_i \mathbf{a}_i} \cdot \mathbf{a}_i$.

Ellipsoid Update

Let $\mathbf{E}_i$ have parameters $\mathbf{Q}_i, \mathbf{c}_i$ and consider the half-ellipse obtained by cutting $\mathbf{E}_i$ with the hyperplane through the center $\mathbf{c}_i$:

$$\mathbf{E}_i \cap \{\mathbf{x} : \mathbf{a}_i^T \mathbf{x} \leq \mathbf{a}_i^T \mathbf{c}_i\}.$$

Key Insight: There is a closed form solution for the smallest ellipse containing a given half-ellipse!

Then $\mathbf{E}_{i+1}$ is the ellipse with parameters:

$$\mathbf{Q}_{i+1} = \frac{d^2}{d^2-1}\left(\mathbf{Q}_i - \frac{2}{d+1}\mathbf{h}\mathbf{h}^T\right) \qquad \mathbf{c}_{i+1} = \mathbf{c}_i - \frac{1}{d+1}\mathbf{h},$$

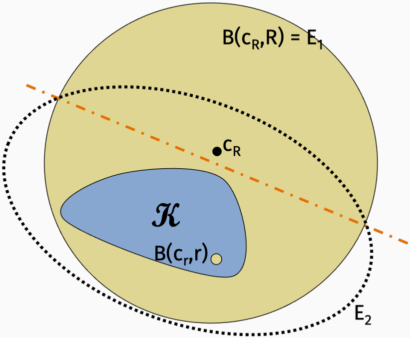
where $\mathbf{h} = \sqrt{\mathbf{a}_i^T \mathbf{Q}_i \mathbf{a}_i} \cdot \mathbf{a}_i$.

Computing the update takes $O(d^2)$ time.

Geometric Observation

Claim: $\text{vol}(E_{i+1}) \leq \left(1 - \frac{1}{2d}\right) \text{vol}(E_i)$.

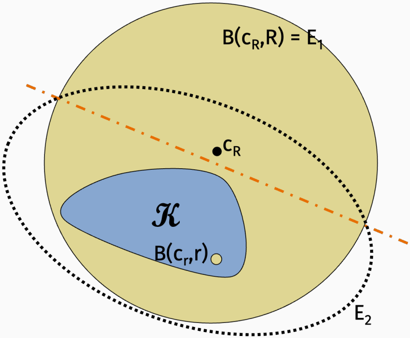
Proof: Via reduction to the "isotropic case". A proof will be posted on the course website.



Geometric Observation

Claim: $\text{vol}(E_{i+1}) \leq \left(1 - \frac{1}{2d}\right) \text{vol}(E_i)$.

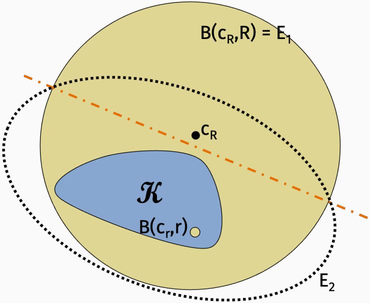
Proof: Via reduction to the "isotropic case". A proof will be posted on the course website.



Not as good as the $(1 - \frac{1}{e})$ constant-factor volume reduction from center-of-gravity, but still very good!

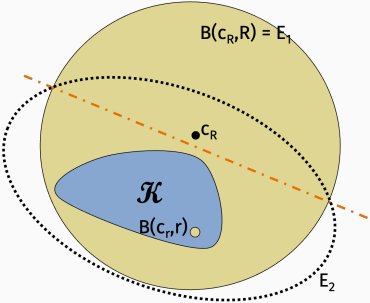
Geometric Observation

Claim: $\mathrm{vol}(E_{i+1}) \leq \left(1 - \frac{1}{2d}\right) \mathrm{vol}(E_i)$



Geometric Observation

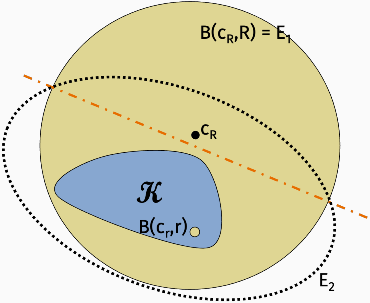
Claim: $\text{vol}(E_{i+1}) \leq \left(1 - \frac{1}{2d}\right) \text{vol}(E_i)$



- After $O(d)$ iterations, we reduce the volume by a constant factor. Why?

Geometric Observation

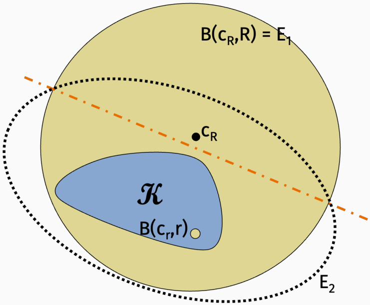
Claim: $\text{vol}(E_{i+1}) \leq \left(1 - \frac{1}{2d}\right) \text{vol}(E_i)$



- After $O(d)$ iterations, we reduce the volume by a constant factor. Why?
- Recall, $\left(1 - \frac{1}{2d}\right)^{2d} \leq 1/e$

Geometric Observation

Claim: $\text{vol}(E_{i+1}) \leq \left(1 - \frac{1}{2d}\right) \text{vol}(E_i)$



- After $O(d)$ iterations, we reduce the volume by a constant factor. Why?
- Recall, $\left(1 - \frac{1}{2d}\right)^{2d} \leq 1/e$
- So how many total iterations do we need?

Geometric Observation

Claim: $\mathrm{vol}(E_{i+1}) \leq \left(1 - \frac{1}{2d}\right) \mathrm{vol}(E_i)$. Recall: $\left(1 - \frac{1}{2d}\right)^{2d} \leq 1/e$.

How many iterations do we need?

Geometric Observation

Claim: $\text{vol}(E_{i+1}) \leq \left(1 - \frac{1}{2d}\right) \text{vol}(E_i)$. Recall: $\left(1 - \frac{1}{2d}\right)^{2d} \leq 1/e$.

How many iterations do we need?

- Start: $\text{vol}(E_1) = \text{vol}(B(\mathbf{c}_R, R)) = R^d$.

Geometric Observation

Claim: $\text{vol}(E_{i+1}) \leq \left(1 - \frac{1}{2d}\right) \text{vol}(E_i)$. Recall: $\left(1 - \frac{1}{2d}\right)^{2d} \leq 1/e$.

How many iterations do we need?

- Start: $\text{vol}(E_1) = \text{vol}(B(\mathbf{c}_R, R)) = R^d$.
- Goal: $\text{vol}(E_T) \leq \text{vol}(B(\mathbf{c}_r, r)) = r^d$.

Geometric Observation

Claim: $\text{vol}(E_{i+1}) \le \left(1 - \frac{1}{2d}\right)\text{vol}(E_i)$. Recall: $\left(1 - \frac{1}{2d}\right)^{2d} \le 1/e$.

How many iterations do we need?

- Start: $\text{vol}(E_1) = \text{vol}(B(\mathbf{c}_R, R)) = R^d$.
- Goal: $\text{vol}(E_T) \le \text{vol}(B(\mathbf{c}_r, r)) = r^d$.
- Need T such that $\left(1 - \frac{1}{2d}\right)^T R^d \le r^d$. It suffices to solve $(1/e)^{T/2d} R^d \le r^d$. Rearranging:

$$e^{T/2d} \ge \left(\frac{R}{r}\right)^d$$
$$\implies T \ge 2d^2 \log(R/r)$$

Geometric Observation

Claim: $\mathrm{vol}(E_{i+1}) \leq \left(1 - \frac{1}{2d}\right) \mathrm{vol}(E_i)$. Recall: $\left(1 - \frac{1}{2d}\right)^{2d} \leq 1/e$.

How many iterations do we need?

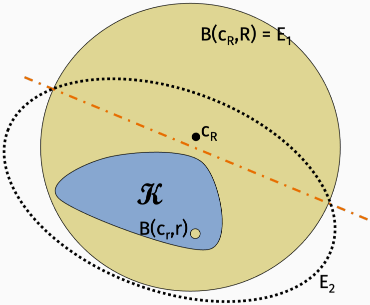
- Start: $\mathrm{vol}(E_1) = \mathrm{vol}(B(\mathbf{c}_R, R)) = R^d$.
- Goal: $\mathrm{vol}(E_T) \leq \mathrm{vol}(B(\mathbf{c}_r, r)) = r^d$.
- Need T such that $\left(1 - \frac{1}{2d}\right)^T R^d \leq r^d$. It suffices to solve $(1/e)^{T/2d} R^d \leq r^d$. Rearranging:

$$e^{T/2d} \geq \left(\frac{R}{r}\right)^d$$
$$\implies T \geq 2d^2 \log(R/r)$$

- Therefore $T = O(d^2 \log(R/r))$ iterations suffice.

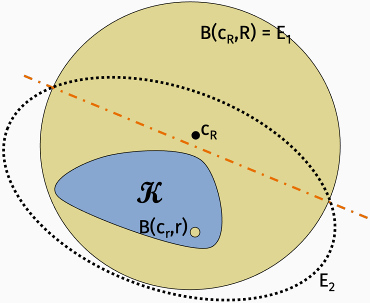
Geometric Observation

Claim: $\mathrm{vol}(E_{i+1}) \leq \left(1 - \frac{1}{2d}\right) \mathrm{vol}(E_i)$



Geometric Observation

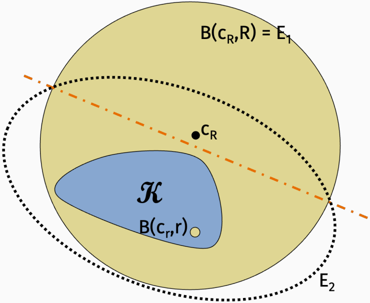
Claim: $\mathrm{vol}(E_{i+1}) \leq \left(1 - \frac{1}{2d}\right) \mathrm{vol}(E_i)$



- After $O(d)$ iterations, we reduce the volume by a constant factor.

Geometric Observation

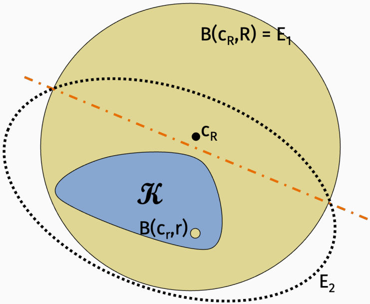
Claim: $\text{vol}(E_{i+1}) \leq \left(1 - \frac{1}{2d}\right) \text{vol}(E_i)$



- After $O(d)$ iterations, we reduce the volume by a constant factor.
- In total we require $O(d^2 \log(R/r))$ iterations to solve the problem.

Geometric Observation

Claim: $\text{vol}(E_{i+1}) \leq \left(1 - \frac{1}{2d}\right) \text{vol}(E_i)$

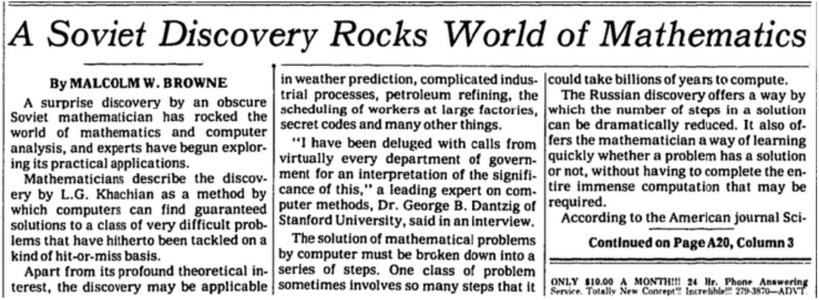


- After $O(d)$ iterations, we reduce the volume by a constant factor.
- In total we require $O(d^2 \log(R/r))$ iterations to solve the problem.
- Complexity for solving $\min_{\mathbf{x} \in \mathcal{S}} f(\mathbf{x})$ is roughly $\tilde{O}(d^4 \log(R/\epsilon))$, hiding logarithmic factors.

Killer Application of Ellipsoid : Linear Programming

Theorem (Khachiyan, 1979)

Assume $n = d$. The ellipsoid method solves any linear program with L-bit integer valued constraints exactly in $O(n^4 L)$ time.



A Soviet Discovery Rocks World of Mathematics

By MALCOLM W. BROWNE

A surprise discovery by an obscure Soviet mathematician has rocked the world of mathematics and computer analysis, and experts have begun exploring its practical applications.

Mathematicians describe the discovery by L.G. Khachian as a method by which computers can find guaranteed solutions to a class of very difficult problems that have hitherto been tackled on a kind of hit-or-miss basis.

Apart from its profound theoretical interest, the discovery may be applicable in weather prediction, complicated industrial processes, petroleum refining, the scheduling of workers at large factories, secret codes and many other things.

"I have been deluged with calls from virtually every department of government for an interpretation of the significance of this," a leading expert on computer methods, Dr. George B. Dantzig of Stanford University, said in an interview.

The solution of mathematical problems by computer must be broken down into a series of steps. One class of problem sometimes involves so many steps that it could take billions of years to compute.

The Russian discovery offers a way by which the number of steps in a solution can be dramatically reduced. It also offers the mathematician a way of learning quickly whether a problem has a solution or not, without having to complete the entire immense computation that may be required.

According to the American journal Sci-

Continued on Page A20, Column 3

Front page of New York Times, November 9, 1979.