

CS675: Convex and Combinatorial Optimization  
Fall 2016  
Consequences of the Ellipsoid Algorithm

Instructor: Shaddin Dughmi

# Outline

- 1 Recapping the Ellipsoid Method
- 2 Complexity of Convex Optimization
- 3 Complexity of Linear Programming
- 4 Equivalence of Separation and Optimization

# Recall: Feasibility Problem

The ellipsoid method solves the following problem.

## Convex Feasibility Problem

Given as input the following

- A description of a compact convex set  $K \subseteq \mathbb{R}^n$
- An ellipsoid  $E(c, Q)$  (typically a ball) containing  $K$
- A rational number  $R > 0$  satisfying  $\text{vol}(E) \leq R$ .
- A rational number  $r > 0$  such that if  $K$  is nonempty, then  $\text{vol}(K) \geq r$ .

Find a point  $x \in K$  or declare that  $K$  is empty.

- Equivalent variant: drop the requirement on volume  $\text{vol}(K)$ , and either find a point  $x \in K$  or an ellipsoid  $E \supseteq K$  with  $\text{vol}(E) < r$ .

All the ellipsoid method needed was the following subroutine

## Separation oracle

An algorithm that takes as input  $x \in \mathbb{R}^n$ , and either certifies  $x \in K$  or outputs a hyperplane separating  $x$  from  $K$ .

- i.e. a vector  $h \in \mathbb{R}^n$  with  $h^\top x \geq h^\top y$  for all  $y \in K$ .
- Equivalently,  $K$  is contained in the halfspace

$$H(h, x) = \{y : h^\top y \leq h^\top x\}$$

with  $x$  at its boundary.

All the ellipsoid method needed was the following subroutine

## Separation oracle

An algorithm that takes as input  $x \in \mathbb{R}^n$ , and either certifies  $x \in K$  or outputs a hyperplane separating  $x$  from  $K$ .

- i.e. a vector  $h \in \mathbb{R}^n$  with  $h^\top x \geq h^\top y$  for all  $y \in K$ .
- Equivalently,  $K$  is contained in the halfspace

$$H(h, x) = \{y : h^\top y \leq h^\top x\}$$

with  $x$  at its boundary.

Examples:

All the ellipsoid method needed was the following subroutine

## Separation oracle

An algorithm that takes as input  $x \in \mathbb{R}^n$ , and either certifies  $x \in K$  or outputs a hyperplane separating  $x$  from  $K$ .

- i.e. a vector  $h \in \mathbb{R}^n$  with  $h^\top x \geq h^\top y$  for all  $y \in K$ .
- Equivalently,  $K$  is contained in the halfspace

$$H(h, x) = \{y : h^\top y \leq h^\top x\}$$

with  $x$  at its boundary.

Examples:

- Explicitly written polytope  $Ay \leq b$ : take  $h = a_i$  to the row of  $A$  corresponding to a constraint violated by  $x$ .

All the ellipsoid method needed was the following subroutine

## Separation oracle

An algorithm that takes as input  $x \in \mathbb{R}^n$ , and either certifies  $x \in K$  or outputs a hyperplane separating  $x$  from  $K$ .

- i.e. a vector  $h \in \mathbb{R}^n$  with  $h^\top x \geq h^\top y$  for all  $y \in K$ .
- Equivalently,  $K$  is contained in the halfspace

$$H(h, x) = \{y : h^\top y \leq h^\top x\}$$

with  $x$  at its boundary.

Examples:

- Explicitly written polytope  $Ay \leq b$ : take  $h = a_i$  to the row of  $A$  corresponding to a constraint violated by  $x$ .
- Convex set given by a family of convex inequalities  $f_i(y) \leq 0$ : Let  $h = \nabla f_i(x)$  for some violated constraint.

All the ellipsoid method needed was the following subroutine

## Separation oracle

An algorithm that takes as input  $x \in \mathbb{R}^n$ , and either certifies  $x \in K$  or outputs a hyperplane separating  $x$  from  $K$ .

- i.e. a vector  $h \in \mathbb{R}^n$  with  $h^\top x \geq h^\top y$  for all  $y \in K$ .
- Equivalently,  $K$  is contained in the halfspace

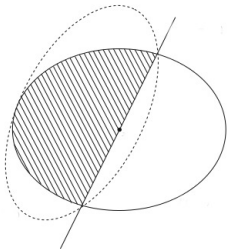
$$H(h, x) = \{y : h^\top y \leq h^\top x\}$$

with  $x$  at its boundary.

Examples:

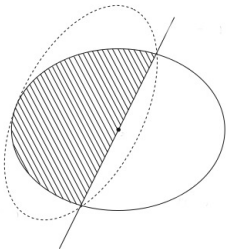
- Explicitly written polytope  $Ay \leq b$ : take  $h = a_i$  to the row of  $A$  corresponding to a constraint violated by  $x$ .
- Convex set given by a family of convex inequalities  $f_i(y) \leq 0$ : Let  $h = \nabla f_i(x)$  for some violated constraint.
- The positive semi-definite cone  $S_n^+$ : Let  $H$  be the outer product  $vv^\top$  of an eigenvector  $v$  of  $X$  corresponding to a negative eigenvalue.





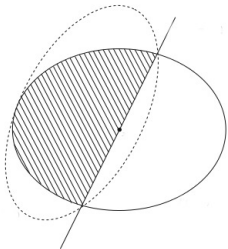
## Ellipsoid Method

- 1 Start with initial ellipsoid  $E = E(c, Q) \supseteq K$
- 2 Using the separation oracle, check if the center  $c \in K$ .
  - If so, terminate and output  $c$ .
  - Otherwise, we get a separating hyperplane  $h$  such that  $K$  is contained in the half-ellipsoid  $E \cap \{y : h^\top y \leq h^\top c\}$
- 3 Let  $E' = E(c', Q')$  be the minimum volume ellipsoid containing the half ellipsoid above.
- 4 If  $\text{vol}(E') \geq r$  then set  $E = E'$  and repeat (step 2), otherwise stop and return “empty”.



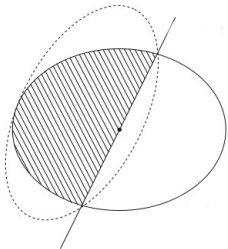
## Ellipsoid Method

- 1 Start with initial ellipsoid  $E = E(c, Q) \supseteq K$
- 2 **Using the separation oracle, check if the center  $c \in K$ .**
  - If so, terminate and output  $c$ .
  - Otherwise, we get a separating hyperplane  $h$  such that  $K$  is contained in the half-ellipsoid  $E \cap \{y : h^\top y \leq h^\top c\}$
- 3 Let  $E' = E(c', Q')$  be the minimum volume ellipsoid containing the half ellipsoid above.
- 4 If  $\text{vol}(E') \geq r$  then set  $E = E'$  and repeat (step 2), otherwise stop and return “empty”.



## Ellipsoid Method

- 1 Start with initial ellipsoid  $E = E(c, Q) \supseteq K$
- 2 Using the separation oracle, check if the center  $c \in K$ .
  - If so, terminate and output  $c$ .
  - Otherwise, we get a separating hyperplane  $h$  such that  $K$  is contained in the half-ellipsoid  $E \cap \{y : h^\top y \leq h^\top c\}$
- 3 **Let  $E' = E(c', Q')$  be the minimum volume ellipsoid containing the half ellipsoid above.**
- 4 If  $\text{vol}(E') \geq r$  then set  $E = E'$  and repeat (step 2), otherwise stop and return “empty”.



## Ellipsoid Method

- 1 Start with initial ellipsoid  $E = E(c, Q) \supseteq K$
- 2 Using the separation oracle, check if the center  $c \in K$ .
  - If so, terminate and output  $c$ .
  - Otherwise, we get a separating hyperplane  $h$  such that  $K$  is contained in the half-ellipsoid  $E \cap \{y : h^\top y \leq h^\top c\}$
- 3 Let  $E' = E(c', Q')$  be the minimum volume ellipsoid containing the half ellipsoid above.
- 4 If  $\text{vol}(E') \geq r$  then set  $E = E'$  and repeat (step 2), otherwise stop and return “empty”.

Using  $T$  to denote the runtime of the separation oracle

## Theorem

The ellipsoid algorithm terminates in time polynomial  $n$ ,  $\ln \frac{R}{r}$ , and  $T$ , and either outputs  $x \in K$  or correctly declares that  $K$  is empty.

We proved most of this (modulo the ellipsoid updating Lemma which we cited and briefly discussed).

Using  $T$  to denote the runtime of the separation oracle

## Theorem

The ellipsoid algorithm terminates in time polynomial  $n$ ,  $\ln \frac{R}{r}$ , and  $T$ , and either outputs  $x \in K$  or correctly declares that  $K$  is empty.

We proved most of this (modulo the ellipsoid updating Lemma which we cited and briefly discussed).

## Note

For runtime polynomial in input size we need

- $T$  polynomial in input size
- $\frac{R}{r}$  exponential in input size

# Outline

- 1 Recapping the Ellipsoid Method
- 2 Complexity of Convex Optimization**
- 3 Complexity of Linear Programming
- 4 Equivalence of Separation and Optimization

## Recall: Convex Optimization Problem

A problem of minimizing a convex function (or maximizing a concave function) over a convex set.

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathcal{X} \end{array}$$

Where  $\mathcal{X} \subseteq \mathbb{R}^n$  is convex and closed, and  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is convex



## Recall: Convex Optimization Problem

A problem of minimizing a convex function (or maximizing a concave function) over a convex set.

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathcal{X} \end{array}$$

Where  $\mathcal{X} \subseteq \mathbb{R}^n$  is convex and closed, and  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is convex

- Recall: A problem  $\Pi$  is a family of **instances**  $I = (f, \mathcal{X})$
- When represented explicitly, often given in **standard form**

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g_i(x) \leq 0, \quad \text{for } i \in \mathcal{C}_1. \\ & a_i^\top x = b_i, \quad \text{for } i \in \mathcal{C}_2. \end{array}$$

- The functions  $f, \{g_i\}_i$  are given in some parametric form allowing evaluation of each function and its derivatives.

## Recall: Convex Optimization Problem

A problem of minimizing a convex function (or maximizing a concave function) over a convex set.

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathcal{X} \end{array}$$

Where  $\mathcal{X} \subseteq \mathbb{R}^n$  is convex and closed, and  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is convex

- We will abstract away details of how instances of a problem are represented, but denote the length of the description by  $\langle I \rangle$
- Require polynomial time (in  $\langle I \rangle$  and  $n$ ) implementation of separation oracle, and other subroutines.

# Solvability of Convex Optimization

There are many subtly different “solvability statements”. This one is the most useful, yet simple to describe, IMO.

## Requirements

We say an algorithm **weakly solves** a convex optimization problem in **polynomial time** if it:

- Takes an approximation parameter  $\epsilon > 0$
- Terminates in time  $\text{poly}(\langle I \rangle, n, \log(\frac{1}{\epsilon}))$
- Returns an  **$\epsilon$ -optimal**  $x \in \mathcal{X}$ :

$$f(x) \leq \min_{y \in \mathcal{X}} f(y) + \epsilon [\max_{y \in \mathcal{X}} f(y) - \min_{y \in \mathcal{X}} f(y)]$$

## Theorem (Polynomial Solvability of CP)

Consider a family  $\Pi$  of convex optimization problems  $I = (f, \mathcal{X})$  admitting the following operations in polynomial time (in  $\langle I \rangle$  and  $n$ ):

- A **separation oracle** for the feasible set  $\mathcal{X} \subseteq \mathbb{R}^n$
- A **first order oracle** for  $f$ : evaluates  $f(x)$  and  $\nabla f(x)$ .
- An algorithm which **computes a starting ellipsoid**  $E \supseteq \mathcal{X}$  with  $\frac{\text{vol}(E)}{\text{vol}(\mathcal{X})} = O(\exp(\langle I \rangle, n))$ .

Then there is a polynomial time algorithm which weakly solves  $\Pi$ .

## Theorem (Polynomial Solvability of CP)

Consider a family  $\Pi$  of convex optimization problems  $I = (f, \mathcal{X})$  admitting the following operations in polynomial time (in  $\langle I \rangle$  and  $n$ ):

- A **separation oracle** for the feasible set  $\mathcal{X} \subseteq \mathbb{R}^n$
- A **first order oracle** for  $f$ : evaluates  $f(x)$  and  $\nabla f(x)$ .
- An algorithm which **computes a starting ellipsoid**  $E \supseteq \mathcal{X}$  with  $\frac{\text{vol}(E)}{\text{vol}(\mathcal{X})} = O(\exp(\langle I \rangle, n))$ .

Then there is a polynomial time algorithm which weakly solves  $\Pi$ .

Let's now prove this, by reducing to the ellipsoid method

## Simplifying Assumption

Assume we are given  $\min_{y \in \mathcal{X}} f(y)$  and  $\max_{y \in \mathcal{X}} f(y)$ . Without loss of generality assume they are  $[0, 1]$ .

## Simplifying Assumption

Assume we are given  $\min_{y \in \mathcal{X}} f(y)$  and  $\max_{y \in \mathcal{X}} f(y)$ . Without loss of generality assume they are  $[0, 1]$ .

Our task reduces to the following convex feasibility problem:

$$\begin{array}{ll} \text{find} & x \\ \text{subject to} & x \in \mathcal{X} \\ & f(x) \leq \epsilon \end{array}$$

## Simplifying Assumption

Assume we are given  $\min_{y \in \mathcal{X}} f(y)$  and  $\max_{y \in \mathcal{X}} f(y)$ . Without loss of generality assume they are  $[0, 1]$ .

Our task reduces to the following convex feasibility problem:

$$\begin{aligned} & \text{find} && x \\ & \text{subject to} && x \in \mathcal{X} \\ & && f(x) \leq \epsilon \end{aligned}$$

We can feed this into the Ellipsoid method!

## Needed Ingredients

- 1 Separation oracle for new feasible set  $K$ :
- 2 Ellipsoid  $E$  containing  $K$ :
- 3 Guarantee that  $\frac{\text{vol}(E)}{\text{vol}(K)} \leq \exp(n, \langle I \rangle, \log \frac{1}{\epsilon})$ :



## Simplifying Assumption

Assume we are given  $\min_{y \in \mathcal{X}} f(y)$  and  $\max_{y \in \mathcal{X}} f(y)$ . Without loss of generality assume they are  $[0, 1]$ .

Our task reduces to the following convex feasibility problem:

$$\begin{array}{ll} \text{find} & x \\ \text{subject to} & x \in \mathcal{X} \\ & f(x) \leq \epsilon \end{array}$$

We can feed this into the Ellipsoid method!

## Needed Ingredients

- 1 Separation oracle for new feasible set  $K$ : Use the separation oracle for  $\mathcal{X}$  and first order oracle for  $f$
- 2 Ellipsoid  $E$  containing  $K$ :
- 3 Guarantee that  $\frac{\text{vol}(E)}{\text{vol}(K)} \leq \exp(n, \langle I \rangle, \log \frac{1}{\epsilon})$ :

## Simplifying Assumption

Assume we are given  $\min_{y \in \mathcal{X}} f(y)$  and  $\max_{y \in \mathcal{X}} f(y)$ . Without loss of generality assume they are  $[0, 1]$ .

Our task reduces to the following convex feasibility problem:

$$\begin{aligned} \text{find} \quad & x \\ \text{subject to} \quad & x \in \mathcal{X} \\ & f(x) \leq \epsilon \end{aligned}$$

We can feed this into the Ellipsoid method!

## Needed Ingredients

- 1 Separation oracle for new feasible set  $K$ : Use the separation oracle for  $\mathcal{X}$  and first order oracle for  $f$
- 2 Ellipsoid  $E$  containing  $K$ : Use the ellipsoid containing  $\mathcal{X}$
- 3 Guarantee that  $\frac{\text{vol}(E)}{\text{vol}(K)} \leq \exp(n, \langle I \rangle, \log \frac{1}{\epsilon})$ :

## Simplifying Assumption

Assume we are given  $\min_{y \in \mathcal{X}} f(y)$  and  $\max_{y \in \mathcal{X}} f(y)$ . Without loss of generality assume they are  $[0, 1]$ .

Our task reduces to the following convex feasibility problem:

$$\begin{array}{ll} \text{find} & x \\ \text{subject to} & x \in \mathcal{X} \\ & f(x) \leq \epsilon \end{array}$$

We can feed this into the Ellipsoid method!

## Needed Ingredients

- 1 Separation oracle for new feasible set  $K$ : Use the separation oracle for  $\mathcal{X}$  and first order oracle for  $f$
- 2 Ellipsoid  $E$  containing  $K$ : Use the ellipsoid containing  $\mathcal{X}$
- 3 Guarantee that  $\frac{\text{vol}(E)}{\text{vol}(K)} \leq \exp(n, \langle I \rangle, \log \frac{1}{\epsilon})$ : Not obvious, but true!

$$K = \{x \in \mathcal{X} : f(x) \leq \epsilon\}$$

## Lemma

$$\text{vol}(K) \geq \epsilon^n \text{vol}(X).$$

This shows that  $\text{vol}(K)$  is only exponentially smaller (in  $n$  and  $\log \frac{1}{\epsilon}$ ) than  $\text{vol}(\mathcal{X})$ , and therefore also  $\text{vol}(E)$ , so it suffices.

$$K = \{x \in \mathcal{X} : f(x) \leq \epsilon\}$$

## Lemma

$$\text{vol}(K) \geq \epsilon^n \text{vol}(X).$$

This shows that  $\text{vol}(K)$  is only exponentially smaller (in  $n$  and  $\log \frac{1}{\epsilon}$ ) than  $\text{vol}(\mathcal{X})$ , and therefore also  $\text{vol}(E)$ , so it suffices.

- Assume wlog  $0 \in \mathcal{X}$  and  $f(0) = \min_{x \in \mathcal{X}} f(x) = 0$ .

$$K = \{x \in \mathcal{X} : f(x) \leq \epsilon\}$$

## Lemma

$$\text{vol}(K) \geq \epsilon^n \text{vol}(X).$$

This shows that  $\text{vol}(K)$  is only exponentially smaller (in  $n$  and  $\log \frac{1}{\epsilon}$ ) than  $\text{vol}(X)$ , and therefore also  $\text{vol}(E)$ , so it suffices.

- Assume wlog  $0 \in \mathcal{X}$  and  $f(0) = \min_{x \in \mathcal{X}} f(x) = 0$ .
- Consider scaling  $\mathcal{X}$  by  $\epsilon$  to get  $\epsilon\mathcal{X}$ .
- $\text{vol}(\epsilon\mathcal{X}) = \epsilon^n \text{vol}(X)$ .

$$K = \{x \in \mathcal{X} : f(x) \leq \epsilon\}$$

## Lemma

$$\text{vol}(K) \geq \epsilon^n \text{vol}(X).$$

This shows that  $\text{vol}(K)$  is only exponentially smaller (in  $n$  and  $\log \frac{1}{\epsilon}$ ) than  $\text{vol}(\mathcal{X})$ , and therefore also  $\text{vol}(E)$ , so it suffices.

- Assume wlog  $0 \in \mathcal{X}$  and  $f(0) = \min_{x \in \mathcal{X}} f(x) = 0$ .
- Consider scaling  $\mathcal{X}$  by  $\epsilon$  to get  $\epsilon\mathcal{X}$ .
- $\text{vol}(\epsilon\mathcal{X}) = \epsilon^n \text{vol}(X)$ .
- We show that  $\epsilon\mathcal{X} \subseteq K$  by showing  $f(y) \leq \epsilon$  for all  $y \in \epsilon\mathcal{X}$ .

$$K = \{x \in \mathcal{X} : f(x) \leq \epsilon\}$$

## Lemma

$$\text{vol}(K) \geq \epsilon^n \text{vol}(X).$$

This shows that  $\text{vol}(K)$  is only exponentially smaller (in  $n$  and  $\log \frac{1}{\epsilon}$ ) than  $\text{vol}(\mathcal{X})$ , and therefore also  $\text{vol}(E)$ , so it suffices.

- Assume wlog  $0 \in \mathcal{X}$  and  $f(0) = \min_{x \in \mathcal{X}} f(x) = 0$ .
- Consider scaling  $\mathcal{X}$  by  $\epsilon$  to get  $\epsilon\mathcal{X}$ .
- $\text{vol}(\epsilon\mathcal{X}) = \epsilon^n \text{vol}(X)$ .
- We show that  $\epsilon\mathcal{X} \subseteq K$  by showing  $f(y) \leq \epsilon$  for all  $y \in \epsilon\mathcal{X}$ .
- Let  $y = \epsilon x$  for  $x \in \mathcal{X}$ , and invoke Jensen's inequality

$$f(y) = f(\epsilon x + (1 - \epsilon)0) \leq \epsilon f(x) + (1 - \epsilon)f(0) \leq \epsilon$$



# Proof (General)

- Denote  $L = \min_{y \in \mathcal{X}} f(y)$  and  $H = \max_{y \in \mathcal{X}} f(y)$
- If we knew the target  $T = L + \epsilon[H - L]$ , we can reduce to solving the feasibility problem over  $K = \{x \in \mathcal{X} : f(x) \leq T\}$ .

# Proof (General)

- Denote  $L = \min_{y \in \mathcal{X}} f(y)$  and  $H = \max_{y \in \mathcal{X}} f(y)$
- If we knew the target  $T = L + \epsilon[H - L]$ , we can reduce to solving the feasibility problem over  $K = \{x \in \mathcal{X} : f(x) \leq T\}$ .
- If we knew it lied in a sufficiently narrow range, we could binary search for  $T$

# Proof (General)

- Denote  $L = \min_{y \in \mathcal{X}} f(y)$  and  $H = \max_{y \in \mathcal{X}} f(y)$
- If we knew the target  $T = L + \epsilon[H - L]$ , we can reduce to solving the feasibility problem over  $K = \{x \in \mathcal{X} : f(x) \leq T\}$ .
- If we knew it lied in a sufficiently narrow range, we could binary search for  $T$
- We don't need to know anything about  $T$ !

## Key Observation

We don't really need to know  $T$ ,  $H$ , or  $L$  to simulate the same execution of the ellipsoid method on  $K$ !!

# Proof (General)

find  $x$   
subject to  $x \in \mathcal{X}$   
 $f(x) \leq T = L + \epsilon[H - L]$

- Simulate the execution of the ellipsoid method on  $K$
- Polynomial number of iterations, terminating with point in  $K$

# Proof (General)

$$\begin{aligned} &\text{find} && x \\ &\text{subject to} && x \in \mathcal{X} \\ &&& f(x) \leq T = L + \epsilon[H - L] \end{aligned}$$

- Simulate the execution of the ellipsoid method on  $K$
- Polynomial number of iterations, terminating with point in  $K$
- Require separation oracle for  $K$  to use  $\nabla f$  only as a last resort
  - This is allowed.
  - Tries to get feasibility whenever possible.

# Proof (General)

find  $x$   
subject to  $x \in \mathcal{X}$   
 $f(x) \leq T = L + \epsilon[H - L]$

- Simulate the execution of the ellipsoid method on  $K$
- Polynomial number of iterations, terminating with point in  $K$
- Require separation oracle for  $K$  to use  $\nabla f$  only as a last resort
  - This is allowed.
  - Tries to get feasibility whenever possible.
- Action of algorithm in each iteration other than the last can be described independently of  $T$ 
  - If ellipsoid center  $c \notin \mathcal{X}$ , use separating hyperplane with  $\mathcal{X}$ .
  - Else use  $\nabla f(c)$

# Proof (General)

$$\begin{aligned} &\text{find} && x \\ &\text{subject to} && x \in \mathcal{X} \\ &&& f(x) \leq T = L + \epsilon[H - L] \end{aligned}$$

- Simulate the execution of the ellipsoid method on  $K$
- Polynomial number of iterations, terminating with point in  $K$
- Require separation oracle for  $K$  to use  $\nabla f$  only as a last resort
  - This is allowed.
  - Tries to get feasibility whenever possible.
- Action of algorithm in each iteration other than the last can be described independently of  $T$ 
  - If ellipsoid center  $c \notin \mathcal{X}$ , use separating hyperplane with  $\mathcal{X}$ .
  - Else use  $\nabla f(c)$
- Run this simulation until enough iterations have passed, and take the best feasible point encountered. This must be in  $K$ .

# Outline

- 1 Recapping the Ellipsoid Method
- 2 Complexity of Convex Optimization
- 3 Complexity of Linear Programming**
- 4 Equivalence of Separation and Optimization



# Recall: Linear Programming

## Recall: Linear Programming Problem

A problem of maximizing a linear function over a polyhedron.

$$\begin{array}{ll} \text{maximize} & c^\top x \\ \text{subject to} & Ax \preceq b \end{array}$$

## Recall: Linear Programming Problem

A problem of maximizing a linear function over a polyhedron.

$$\begin{array}{ll} \text{maximize} & c^\top x \\ \text{subject to} & Ax \preceq b \end{array}$$

- When stated in standard form, optimal solution occurs at a vertex.

## Recall: Linear Programming Problem

A problem of maximizing a linear function over a polyhedron.

$$\begin{array}{ll} \text{maximize} & c^\top x \\ \text{subject to} & Ax \preceq b \end{array}$$

- When stated in standard form, optimal solution occurs at a vertex.
- We will consider both explicitly and implicit LPs
  - Explicit: given by  $A$ ,  $b$  and  $c$
  - Implicit: Given by  $c$  and a separation oracle for  $Ax \leq b$ .

## Recall: Linear Programming Problem

A problem of maximizing a linear function over a polyhedron.

$$\begin{array}{ll} \text{maximize} & c^\top x \\ \text{subject to} & Ax \preceq b \end{array}$$

- When stated in standard form, optimal solution occurs at a vertex.
- We will consider both explicitly and implicit LPs
  - Explicit: given by  $A$ ,  $b$  and  $c$
  - Implicit: Given by  $c$  and a separation oracle for  $Ax \preceq b$ .
- In both cases, we require all numbers to be rational

# Recall: Linear Programming

## Recall: Linear Programming Problem

A problem of maximizing a linear function over a polyhedron.

$$\begin{array}{ll} \text{maximize} & c^\top x \\ \text{subject to} & Ax \preceq b \end{array}$$

- When stated in standard form, optimal solution occurs at a vertex.
- We will consider both explicitly and implicit LPs
  - Explicit: given by  $A$ ,  $b$  and  $c$
  - Implicit: Given by  $c$  and a separation oracle for  $Ax \preceq b$ .
- In both cases, we require all numbers to be rational
- In the explicit case, we require polynomial time in  $\langle A \rangle$ ,  $\langle b \rangle$ , and  $\langle c \rangle$ , the number of bits used to represent the parameters of the LP.

# Recall: Linear Programming

## Recall: Linear Programming Problem

A problem of maximizing a linear function over a polyhedron.

$$\begin{array}{ll} \text{maximize} & c^\top x \\ \text{subject to} & Ax \preceq b \end{array}$$

- When stated in standard form, optimal solution occurs at a vertex.
- We will consider both explicitly and implicit LPs
  - Explicit: given by  $A$ ,  $b$  and  $c$
  - Implicit: Given by  $c$  and a separation oracle for  $Ax \leq b$ .
- In both cases, we require all numbers to be rational
- In the explicit case, we require polynomial time in  $\langle A \rangle$ ,  $\langle b \rangle$ , and  $\langle c \rangle$ , the number of bits used to represent the parameters of the LP.
- In the implicit case, we require polynomial time in the bit complexity of individual entries of  $A, b, c$ .

## Theorem (Polynomial Solvability of Explicit LP)

*There is a polynomial time algorithm for linear programming, when the linear program is represented explicitly.*

## Proof Sketch (Informal)

Using result for weakly solving convex programs, we need 4 things:

- A separation oracle for  $Ax \leq b$ : trivial when explicitly represented
- A first order oracle for  $c^T x$ : also trivial
- A bounding ellipsoid of volume at most an exponential times the volume of the feasible polyhedron: tricky
- A way of “rounding” an  $\epsilon$ -optimal solution to an optimal vertex solution: tricky

## Theorem (Polynomial Solvability of Explicit LP)

*There is a polynomial time algorithm for linear programming, when the linear program is represented explicitly.*

## Proof Sketch (Informal)

Using result for weakly solving convex programs, we need 4 things:

- A separation oracle for  $Ax \leq b$ : trivial when explicitly represented
- A first order oracle for  $c^T x$ : also trivial
- A bounding ellipsoid of volume at most an exponential times the volume of the feasible polyhedron: tricky
- A way of “rounding” an  $\epsilon$ -optimal solution to an optimal vertex solution: tricky

Solution to both issues involves tedious accounting of numerical issues



## Ellipsoid and Volume Bound (Informal)

Key to tackling both difficulties is the following observation:

### Lemma

*Let  $v$  be vertex of the polyhedron  $Ax \leq b$ . It is the case that  $v$  has polynomial bit complexity, i.e.  $\langle v \rangle \leq M$ , where  $M = O(\text{poly}(\langle A \rangle, \langle b \rangle))$ .*

Specifically, the solution of a system of linear equations has bit complexity polynomially related to that of the equations.

# Ellipsoid and Volume Bound (Informal)

Key to tackling both difficulties is the following observation:

## Lemma

*Let  $v$  be vertex of the polyhedron  $Ax \leq b$ . It is the case that  $v$  has polynomial bit complexity, i.e.  $\langle v \rangle \leq M$ , where  $M = O(\text{poly}(\langle A \rangle, \langle b \rangle))$ .*

Specifically, the solution of a system of linear equations has bit complexity polynomially related to that of the equations.

- Bounding ellipsoid: all vertices contained in the box  $-2^M \leq x \leq 2^M$ , which in turn is contained in an ellipsoid of volume exponential in  $M$  and  $n$ .

# Ellipsoid and Volume Bound (Informal)

Key to tackling both difficulties is the following observation:

## Lemma

*Let  $v$  be vertex of the polyhedron  $Ax \leq b$ . It is the case that  $v$  has polynomial bit complexity, i.e.  $\langle v \rangle \leq M$ , where  $M = O(\text{poly}(\langle A \rangle, \langle b \rangle))$ .*

Specifically, the solution of a system of linear equations has bit complexity polynomially related to that of the equations.

- Volume lowerbound when feasible set is full dimensional: follows from bit complexity of vertices.
- More generally, need to instead solve a “relaxed problem”. Specifically, relaxing to  $Ax \leq b + \epsilon$ , for sufficiently small  $\epsilon$  with  $\langle \epsilon \rangle = \text{poly}(M)$ . Gives volume exponentially small in  $M$ , but no smaller. Still close enough to original polyhedron so solution to relaxed problem can be “rounded” to solution of the original problem.

# Ellipsoid and Volume Bound (Informal)

Key to tackling both difficulties is the following observation:

## Lemma

*Let  $v$  be vertex of the polyhedron  $Ax \leq b$ . It is the case that  $v$  has polynomial bit complexity, i.e.  $\langle v \rangle \leq M$ , where  $M = O(\text{poly}(\langle A \rangle, \langle b \rangle))$ .*

Specifically, the solution of a system of linear equations has bit complexity polynomially related to that of the equations.

- Rounding to a vertex: If a point  $y$  is  $\epsilon$ -optimal for the  $\epsilon$ -relaxed problem, for sufficiently small  $\epsilon$  chosen carefully to polynomial in description of input, then rounding to the nearest  $x$  with  $M$  bits recovers the vertex.

## Theorem (Polynomial Solvability of Implicit LP)

Consider a family  $\Pi$  of linear programming problems  $I = (A, b, c)$  admitting the following operations in polynomial time (in  $\langle I \rangle$  and  $n$ ):

- A **separation oracle** for the polyhedron  $Ax \leq b$
- Explicit access to  $c$

Moreover, assume that every  $\langle a_{ij} \rangle$ ,  $\langle b_i \rangle$ ,  $\langle c_j \rangle$  are at most  $\text{poly}(\langle I \rangle, n)$ . Then there is a polynomial time algorithm for  $\Pi$  (both primal and dual\*).

## Informal Proof Sketch (Primal)

## Theorem (Polynomial Solvability of Implicit LP)

Consider a family  $\Pi$  of linear programming problems  $I = (A, b, c)$  admitting the following operations in polynomial time (in  $\langle I \rangle$  and  $n$ ):

- A **separation oracle** for the polyhedron  $Ax \leq b$
- Explicit access to  $c$

Moreover, assume that every  $\langle a_{ij} \rangle$ ,  $\langle b_i \rangle$ ,  $\langle c_j \rangle$  are at most  $\text{poly}(\langle I \rangle, n)$ . Then there is a polynomial time algorithm for  $\Pi$  (both primal and dual\*).

## Informal Proof Sketch (Primal)

- Separation oracle and first order oracle are given

## Theorem (Polynomial Solvability of Implicit LP)

Consider a family  $\Pi$  of linear programming problems  $I = (A, b, c)$  admitting the following operations in polynomial time (in  $\langle I \rangle$  and  $n$ ):

- A **separation oracle** for the polyhedron  $Ax \leq b$
- Explicit access to  $c$

Moreover, assume that every  $\langle a_{ij} \rangle$ ,  $\langle b_i \rangle$ ,  $\langle c_j \rangle$  are at most  $\text{poly}(\langle I \rangle, n)$ . Then there is a polynomial time algorithm for  $\Pi$  (both primal and dual\*).

## Informal Proof Sketch (Primal)

- Separation oracle and first order oracle are given
- Rounding to a vertex exactly as in the explicit case.
  - Every vertex  $v$  still has polynomial bit complexity  $M$

## Theorem (Polynomial Solvability of Implicit LP)

Consider a family  $\Pi$  of linear programming problems  $I = (A, b, c)$  admitting the following operations in polynomial time (in  $\langle I \rangle$  and  $n$ ):

- A **separation oracle** for the polyhedron  $Ax \leq b$
- Explicit access to  $c$

Moreover, assume that every  $\langle a_{ij} \rangle, \langle b_i \rangle, \langle c_j \rangle$  are at most  $\text{poly}(\langle I \rangle, n)$ . Then there is a polynomial time algorithm for  $\Pi$  (both primal and dual\*).

## Informal Proof Sketch (Primal)

- Separation oracle and first order oracle are given
- Rounding to a vertex exactly as in the explicit case.
  - Every vertex  $v$  still has polynomial bit complexity  $M$
- Bounding ellipsoid: Still true that we get a bounding ellipsoid of volume exponential in  $\langle I \rangle$  and  $n$



## Theorem (Polynomial Solvability of Implicit LP)

Consider a family  $\Pi$  of linear programming problems  $I = (A, b, c)$  admitting the following operations in polynomial time (in  $\langle I \rangle$  and  $n$ ):

- A **separation oracle** for the polyhedron  $Ax \leq b$
- Explicit access to  $c$

Moreover, assume that every  $\langle a_{ij} \rangle$ ,  $\langle b_i \rangle$ ,  $\langle c_j \rangle$  are at most  $\text{poly}(\langle I \rangle, n)$ . Then there is a polynomial time algorithm for  $\Pi$  (both primal and dual\*).

## Informal Proof Sketch (Primal)

- Separation oracle and first order oracle are given
- Rounding to a vertex exactly as in the explicit case.
  - Every vertex  $v$  still has polynomial bit complexity  $M$
- Bounding ellipsoid: Still true that we get a bounding ellipsoid of volume exponential in  $\langle I \rangle$  and  $n$
- However, no lowerbound on the volume of  $Ax \leq b$  — can't relax to  $Ax \leq b + \epsilon$  as in the explicit case.

## Theorem (Polynomial Solvability of Implicit LP)

Consider a family  $\Pi$  of linear programming problems  $I = (A, b, c)$  admitting the following operations in polynomial time (in  $\langle I \rangle$  and  $n$ ):

- A **separation oracle** for the polyhedron  $Ax \leq b$
- Explicit access to  $c$

Moreover, assume that every  $\langle a_{ij} \rangle, \langle b_i \rangle, \langle c_j \rangle$  are at most  $\text{poly}(\langle I \rangle, n)$ . Then there is a polynomial time algorithm for  $\Pi$  (both primal and dual\*).

## Informal Proof Sketch (Primal)

- Separation oracle and first order oracle are given
- Rounding to a vertex exactly as in the explicit case.
  - Every vertex  $v$  still has polynomial bit complexity  $M$
- Bounding ellipsoid: Still true that we get a bounding ellipsoid of volume exponential in  $\langle I \rangle$  and  $n$
- However, no lowerbound on the volume of  $Ax \leq b$  — can't relax to  $Ax \leq b + \epsilon$  as in the explicit case.
  - It turns out this is still OK, but takes a lot of work.

## Theorem (Polynomial Solvability of Implicit LP)

Consider a family  $\Pi$  of linear programming problems  $I = (A, b, c)$  admitting the following operations in polynomial time (in  $\langle I \rangle$  and  $n$ ):

- A **separation oracle** for the polyhedron  $Ax \leq b$
- Explicit access to  $c$

Moreover, assume that every  $\langle a_{ij} \rangle, \langle b_i \rangle, \langle c_j \rangle$  are at most  $\text{poly}(\langle I \rangle, n)$ . Then there is a polynomial time algorithm for  $\Pi$  (both primal and dual\*).

For the dual, we need equivalence of separation and optimization. Also, we necessarily get a solution to a normalized version of the dual.  
(HW)

# Outline

- 1 Recapping the Ellipsoid Method
- 2 Complexity of Convex Optimization
- 3 Complexity of Linear Programming
- 4 Equivalence of Separation and Optimization**

# Separation and Optimization

- One interpretation of the previous theorem is that optimization of linear functions over a polytope of polynomial bit complexity reduces to implementing a separation oracle
- As it turns out, the two tasks are polynomial-time equivalent.

# Separation and Optimization

- One interpretation of the previous theorem is that optimization of linear functions over a polytope of polynomial bit complexity reduces to implementing a separation oracle
- As it turns out, the two tasks are polynomial-time equivalent.

Lets formalize the two questions, parametrized by a polytope  $P$ .

## Linear Optimization Problem

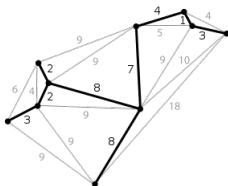
- Input: Linear objective  $c \in \mathbb{R}^n$ .
- Output:  $\operatorname{argmax}_{x \in P} c^\top x$ .

## Separation Problem

- Input:  $y \in \mathbb{R}^n$
- Output: Decide that  $y \in P$ , or else find  $h \in \mathbb{R}^n$  s.t.  $h^\top x < h^\top y$  for all  $x \in P$ .

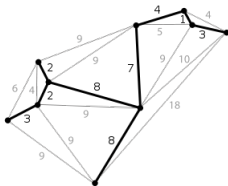
# Recall: Minimum Cost Spanning Tree

Given a connected undirected graph  $G = (V, E)$ , and costs  $c_e$  on edges  $e$ , find a minimum cost spanning tree of  $G$ .



# Recall: Minimum Cost Spanning Tree

Given a connected undirected graph  $G = (V, E)$ , and costs  $c_e$  on edges  $e$ , find a minimum cost spanning tree of  $G$ .



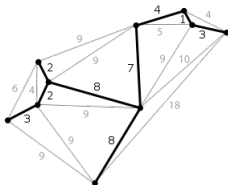
## Spanning Tree Polytope

$$\begin{aligned} \sum_{e \subseteq X} x_e &\leq |X| - 1, & \text{for } X \subset V. \\ \sum_{e \in E} x_e &= n - 1 \\ x_e &\geq 0, & \text{for } e \in E. \end{aligned}$$



# Recall: Minimum Cost Spanning Tree

Given a connected undirected graph  $G = (V, E)$ , and costs  $c_e$  on edges  $e$ , find a minimum cost spanning tree of  $G$ .



## Spanning Tree Polytope

$$\begin{aligned} \sum_{e \subseteq X} x_e &\leq |X| - 1, & \text{for } X \subset V. \\ \sum_{e \in E} x_e &= n - 1 \\ x_e &\geq 0, & \text{for } e \in E. \end{aligned}$$

- Optimization: Find the minimum/maximum weight spanning tree
- Separation: Find  $X \subset V$  with  $\sum_{e \subseteq X} x_e > |X| - 1$ , if one exists
  - i.e. When edge weights are  $x$ , find a “dense” subgraph

## Theorem (Equivalence of Separation and Optimization for Polytopes)

*Consider a family  $\mathcal{P}$  of polytopes  $P = \{x : Ax \leq b\}$  described implicitly using  $\langle P \rangle$  bits, and satisfying  $\langle a_{ij} \rangle, \langle b_i \rangle \leq \text{poly}(\langle P \rangle, n)$ . Then the separation problem is solvable in  $\text{poly}(\langle P \rangle, n, \langle y \rangle)$  time for  $P \in \mathcal{P}$  if and only if the linear optimization problem is solvable in  $\text{poly}(\langle P \rangle, n, \langle c \rangle)$  time.*

- Colloquially, we say such polytope families are **solvable**.

## Theorem (Equivalence of Separation and Optimization for Polytopes)

*Consider a family  $\mathcal{P}$  of polytopes  $P = \{x : Ax \leq b\}$  described implicitly using  $\langle P \rangle$  bits, and satisfying  $\langle a_{ij} \rangle, \langle b_i \rangle \leq \text{poly}(\langle P \rangle, n)$ . Then the separation problem is solvable in  $\text{poly}(\langle P \rangle, n, \langle y \rangle)$  time for  $P \in \mathcal{P}$  if and only if the linear optimization problem is solvable in  $\text{poly}(\langle P \rangle, n, \langle c \rangle)$  time.*

- Colloquially, we say such polytope families are **solvable**.
- E.g. Spanning tree polytopes, represented by graphs, are solvable.

## Theorem (Equivalence of Separation and Optimization for Polytopes)

*Consider a family  $\mathcal{P}$  of polytopes  $P = \{x : Ax \leq b\}$  described implicitly using  $\langle P \rangle$  bits, and satisfying  $\langle a_{ij} \rangle, \langle b_i \rangle \leq \text{poly}(\langle P \rangle, n)$ . Then the separation problem is solvable in  $\text{poly}(\langle P \rangle, n, \langle y \rangle)$  time for  $P \in \mathcal{P}$  if and only if the linear optimization problem is solvable in  $\text{poly}(\langle P \rangle, n, \langle c \rangle)$  time.*

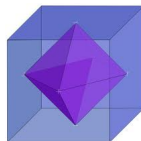
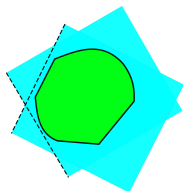
- Colloquially, we say such polytope families are **solvable**.
- E.g. Spanning tree polytopes, represented by graphs, are solvable.
- We already sketched the proof of the forward direction
  - Separation  $\Rightarrow$  optimization

## Theorem (Equivalence of Separation and Optimization for Polytopes)

*Consider a family  $\mathcal{P}$  of polytopes  $P = \{x : Ax \leq b\}$  described implicitly using  $\langle P \rangle$  bits, and satisfying  $\langle a_{ij} \rangle, \langle b_i \rangle \leq \text{poly}(\langle P \rangle, n)$ . Then the separation problem is solvable in  $\text{poly}(\langle P \rangle, n, \langle y \rangle)$  time for  $P \in \mathcal{P}$  if and only if the linear optimization problem is solvable in  $\text{poly}(\langle P \rangle, n, \langle c \rangle)$  time.*

- Colloquially, we say such polytope families are **solvable**.
- E.g. Spanning tree polytopes, represented by graphs, are solvable.
- We already sketched the proof of the forward direction
  - Separation  $\Rightarrow$  optimization
- For the other direction, we need **polars**

# Recall: Polar Duality of Convex Sets



One way of representing the all halfspaces containing a convex set.

## Polar

Let  $S \subseteq \mathbb{R}^n$  be a closed convex set containing the origin. The **polar** of  $S$  is defined as follows:

$$S^\circ = \{y : x \cdot y \leq 1 \text{ for all } x \in S\}$$

## Note

- Every halfspace  $a^\top x \leq b$  with  $b \neq 0$  can be written as a “normalized” inequality  $y^\top x \leq 1$ , by dividing by  $b$ .
- $S^\circ$  can be thought of as the normalized representations of halfspaces containing  $S$ .

## Properties of the Polar

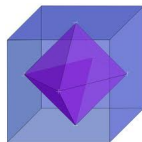
- 1 If  $S$  is bounded and  $0 \in \text{interior}(S)$ , then the same holds for  $S^\circ$ .
- 2  $S^{\circ\circ} = S$



$$S = \{x : y \cdot x \leq 1 \text{ for all } y \in S^\circ\}$$



$$S^\circ = \{y : x \cdot y \leq 1 \text{ for all } x \in S\}$$



## Polytopes

Given a polytope  $P$  represented as  $Ax \preceq \vec{1}$ , the polar  $P^\circ$  is the convex hull of the rows of  $A$ .

- Facets of  $P$  correspond to vertices of  $P^\circ$ .
- Dually, vertices of  $P$  correspond to facets of  $P^\circ$ .



# Proof Outline: Optimization $\Rightarrow$ Separation

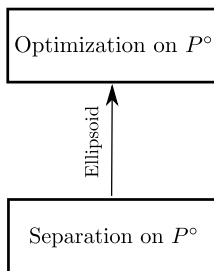
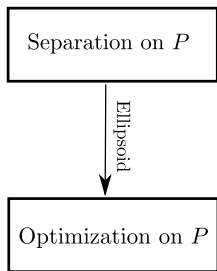


Separation on  $P$

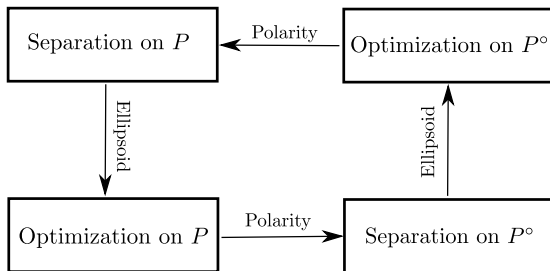
Ellipsoid

Optimization on  $P$

# Proof Outline: Optimization $\Rightarrow$ Separation



# Proof Outline: Optimization $\Rightarrow$ Separation





$$S = \{x : y \cdot x \leq 1 \text{ for all } y \in S^\circ\}$$



$$S^\circ = \{y : x \cdot y \leq 1 \text{ for all } x \in S\}$$



$$S = \{x : y \cdot x \leq 1 \text{ for all } y \in S^\circ\}$$

$$S^\circ = \{y : x \cdot y \leq 1 \text{ for all } x \in S\}$$

## Lemma

Separation over  $S$  reduces in constant time to optimization over  $S^\circ$ , and vice versa since  $S^{\circ\circ} = S$ .



$$S = \{x : y \cdot x \leq 1 \text{ for all } y \in S^\circ\}$$

$$S^\circ = \{y : x \cdot y \leq 1 \text{ for all } x \in S\}$$

## Lemma

Separation over  $S$  reduces in constant time to optimization over  $S^\circ$ , and vice versa since  $S^{\circ\circ} = S$ .

## Proof

- We are given vector  $x$ , and must check whether  $x \in S$ , and if not output separating hyperplane.



$$S = \{x : y \cdot x \leq 1 \text{ for all } y \in S^\circ\}$$

$$S^\circ = \{y : x \cdot y \leq 1 \text{ for all } x \in S\}$$

## Lemma

Separation over  $S$  reduces in constant time to optimization over  $S^\circ$ , and vice versa since  $S^{\circ\circ} = S$ .

## Proof

- We are given vector  $x$ , and must check whether  $x \in S$ , and if not output separating hyperplane.
- $x \in S$  iff  $y \cdot x \leq 1$  for all  $y \in S^\circ$



$$S = \{x : y \cdot x \leq 1 \text{ for all } y \in S^\circ\}$$

$$S^\circ = \{y : x \cdot y \leq 1 \text{ for all } x \in S\}$$

## Lemma

Separation over  $S$  reduces in constant time to optimization over  $S^\circ$ , and vice versa since  $S^{\circ\circ} = S$ .

## Proof

- We are given vector  $x$ , and must check whether  $x \in S$ , and if not output separating hyperplane.
- $x \in S$  iff  $y \cdot x \leq 1$  for all  $y \in S^\circ$
- equivalently, iff  $\max_{y \in S^\circ} y \cdot x \leq 1$ .





$$S = \{x : y \cdot x \leq 1 \text{ for all } y \in S^\circ\}$$

$$S^\circ = \{y : x \cdot y \leq 1 \text{ for all } x \in S\}$$

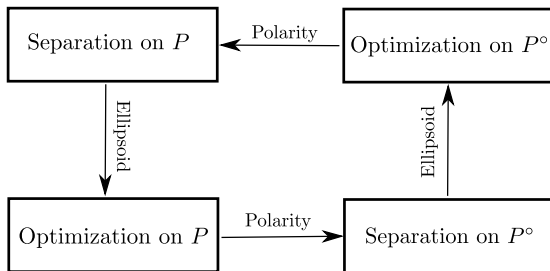
## Lemma

Separation over  $S$  reduces in constant time to optimization over  $S^\circ$ , and vice versa since  $S^{\circ\circ} = S$ .

## Proof

- We are given vector  $x$ , and must check whether  $x \in S$ , and if not output separating hyperplane.
- $x \in S$  iff  $y \cdot x \leq 1$  for all  $y \in S^\circ$
- equivalently, iff  $\max_{y \in S^\circ} y \cdot x \leq 1$ .
- If we find  $y \in S^\circ$  s.t.  $y \cdot x > 1$ , then  $y$  is the separating hyperplane
  - $y^\top z \leq 1 < y^\top x$  for every  $z \in S$ .

# Optimization $\iff$ Separation



# Beyond Polytopes

Essentially everything we proved about equivalence of separation and optimization for polytopes extends (approximately) to arbitrary convex sets.

# Beyond Polytopes

Essentially everything we proved about equivalence of separation and optimization for polytopes extends (approximately) to arbitrary convex sets.

Problems parametrized by  $P$ , a closed convex set.

## Weak Optimization Problem

- Input: Linear objective  $c \in \mathbb{R}^n$ .
- Output:  $x \in P^{+\epsilon}$ , and  $c^\top x \geq \max_{x' \in P} c^\top x' - \epsilon$

## Weak Separation Problem

- Input:  $y \in \mathbb{R}^n$
- Output: Decide that  $y \in P^{-\epsilon}$ , or else find  $h \in \mathbb{R}^n$  with  $\|h\| = 1$  s.t.  $h^\top x < h^\top y + \epsilon$  for all  $x \in P$ .

# Beyond Polytopes

Essentially everything we proved about equivalence of separation and optimization for polytopes extends (approximately) to arbitrary convex sets.

Problems parametrized by  $P$ , a closed convex set.

## Weak Optimization Problem

- Input: Linear objective  $c \in \mathbb{R}^n$ .
- Output:  $x \in P^{+\epsilon}$ , and  $c^\top x \geq \max_{x' \in P} c^\top x' - \epsilon$

## Weak Separation Problem

- Input:  $y \in \mathbb{R}^n$
- Output: Decide that  $y \in P^{-\epsilon}$ , or else find  $h \in \mathbb{R}^n$  with  $\|h\| = 1$  s.t.  $h^\top x < h^\top y + \epsilon$  for all  $x \in P$ .

I could have equivalently stated the weak optimization problem for convex functions instead of linear.

## Theorem (Equivalence of Separation and Optimization for Convex Sets)

*Consider a family  $\mathcal{P}$  of convex sets described implicitly using  $\langle P \rangle$  bits. Then the weak separation problem is solvable in  $\text{poly}(\langle P \rangle, n, \langle y \rangle, \log(1/\epsilon))$  time for  $P \in \mathcal{P}$  if and only if the weak optimization problem is also solvable in  $\text{poly}(\langle P \rangle, n, \langle c \rangle, \log(1/\epsilon))$  time.*

## Theorem (Equivalence of Separation and Optimization for Convex Sets)

*Consider a family  $\mathcal{P}$  of convex sets described implicitly using  $\langle P \rangle$  bits. Then the weak separation problem is solvable in  $\text{poly}(\langle P \rangle, n, \langle y \rangle, \log(1/\epsilon))$  time for  $P \in \mathcal{P}$  if and only if the weak optimization problem is also solvable in  $\text{poly}(\langle P \rangle, n, \langle c \rangle, \log(1/\epsilon))$  time.*

- The “approximation” in this statement is necessary, since we can’t solve convex optimization problems exactly.
- Weak separation suffices for ellipsoid, which is only approximately optimal anyways
- By polarity, weak optimization is equivalent to weak separation

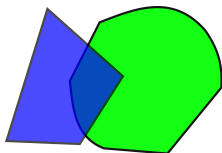
## Theorem (Equivalence of Separation and Optimization for Convex Sets)

*Consider a family  $\mathcal{P}$  of convex sets described implicitly using  $\langle P \rangle$  bits. Then the weak separation problem is solvable in  $\text{poly}(\langle P \rangle, n, \langle y \rangle, \log(1/\epsilon))$  time for  $P \in \mathcal{P}$  if and only if the weak optimization problem is also solvable in  $\text{poly}(\langle P \rangle, n, \langle c \rangle, \log(1/\epsilon))$  time.*

- The “approximation” in this statement is necessary, since we can’t solve convex optimization problems exactly.
- Weak separation suffices for ellipsoid, which is only approximately optimal anyways
- By polarity, weak optimization is equivalent to weak separation
- For proof / details, see the GLS book.



# Implication: Operations preserving solvability

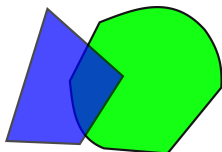


- Assume you can efficiently optimize over two convex sets  $P$  and  $Q$

## Question

What about  $P \cup Q$  and  $P \cap Q$ ?

# Implication: Operations preserving solvability



- Assume you can efficiently optimize over two convex sets  $P$  and  $Q$

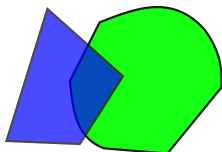
## Question

What about  $P \cup Q$  and  $P \cap Q$ ?

## $P \cup Q$

- Yes! Simply optimize over each separately, and take the better of the two outcomes.
- Equivalent to optimizing over the convex hull of  $P \cup Q$ .
- Implication of Separation/optimization equivalence: there is a separation oracle for  $\text{convexhull}(P \cup Q)$ .

# Implication: Operations preserving solvability



- Assume you can efficiently optimize over two convex sets  $P$  and  $Q$

## Question

What about  $P \cup Q$  and  $P \cap Q$ ?

## $P \cap Q$

- Yes! Follows from equivalence of separation and optimization.
- Specifically, can separate over  $P$  and  $Q$  individually, therefore can separate over  $P \cap Q$ , and then can optimize over  $P \cap Q$ .
- Applications: colorful spanning tree, cardinality-constrained matching, ...

## Problem

Given a point  $x \in \mathcal{P}$ , where  $\mathcal{P} \subseteq \mathbb{R}^n$  is a solvable polytope, write  $x$  as a convex combination of  $n + 1$  vertices of  $\mathcal{P}$ , and do so in polynomial time.

- Existence: Caratheodory's theorem.
- E.g. Birkhoff Von-Neumann, fractional spanning trees, fractional matchings, ...
- Follows from equivalence of separation and optimization. See HW4.